

Jaakko Perttu

TEOLLISUUSROBOTIN KÄYTTÖ KOKOONPANOTYÖN
AUTOMATISOINNISSA

Diplomityö, joka on jätetty opinnäytteenä tarkastettavaksi
diplomi-insinöörin tutkintoa varten Espoossa 25.10.1984.

Työn valvoja



Antti Niemi

Työn ohjaaja



Veikko Mäkipää

14911

Tekijä ja työn nimi : Perttu, Jaakko Juhani

Teollisuusrobotin käyttö kokoonpanotyön automatisoinnissa

Päivämäärä : 25.10.1984

Sivumäärä : 65

Osasto :

Professuuri : 1.74

Sähkötekniillinen osasto

Säätötekniikka

Työn valvoja : professori Antti Niemi

Työn ohjaaja : DI Veikko Mäkipää

Diplomityössä tutkittiin kokoonpanotyön robotisointia. Tavoitteena oli erityisesti testata sensoreiden tarve ja käytettävyys robotin työskentelyssä sekä ohjelmointimenetelmän vaikutus robottisovelluksen toteutukseen.

Työssä kuvataan kokoonpanotyöhön liittyvät käsitteet ja menetelmät. Lisäksi käsitellään olemassa olevan robottiteknologian soveltamista kokoonpanotehtäviin sekä arvioidaan tulevaisuuden kehityssuuntia. Kokoonpanokokeessa käytetyn IBM 7565-robotin ominaisuudet kuvataan yksityiskohtaisesti.

Käytännön kokeessa kalustettiin sähkömoottorin käynnistinkaappi. Kokoonpanon virheettömyyden valvontaan käytettiin sekä kouran tuntosensoreita että ruuvitaltan alipainesensoria. Tuntosensoreita käytettiin komponenttien sovituksessa sekä virhetilojen havaitsemisessa. Alipainesensoria käytettiin ruuvauksen virhetilojen havaitsemisessa. Kokoonpano-ohjelmassa oli useita virheestätoipumisrutiineja, joiden avulla robotti selvitti mahdolliset virhetilanteet ja kykeni näin jatkamaan toimintaansa virheistä huolimatta. Vuorovaikutteista ohjelmointijärjestelmää tarvittiin näiden elpymisrutiinien virittämiseen.

Kokoonpanotehtävä onnistui hyvin ja se täytti kaikki sille ennalta asetetut vaatimukset. Sovelluksen kehittämiseen käytetty aika, noin kuusi viikkoa osoitti, että vastaavat kokoonpanotehtävät voidaan toteuttaa taloudellisesti normaalissa tuotantoympäristössä. Sensoreiden käyttö yhdessä vuorovaikutteisen ohjelmointimenetelmän kanssa olivat välttämättömiä esimerkkisovelluksen toteuttamisessa.

Hakusanat: Robotti, kokoonpano, sensorit

ESIPUHE

Olen tehnyt tämän diplomityön Oy International Business Machines Ab:n toimeksiannosta. Tehtävänäni on ollut tutustua kokoonpanotehtävien robotisointiin ja selvittää käytännön kokeilla IBM 7565-robotin soveltuvuus Oy Wärtsilä Ab:n Turun telakan valmistamien kytkinkaappien kokoonpanoon. Soveltuvuuskokeet tehtiin Ruotsin IBM:n Järfällän tehtaiden robottilaitteistolla.

Työssäni tarkastelen kokoonpanotehtävää prosessina ja selvitän siihen liittyvää ongelmakenttää robotiikan kannalta. Käytännön työssä keskityn sovelluskohtaisten erityisongelmien selvittämiseen ja käyttämieni menetelmien kuvaamiseen.

Kiitän työni tarkastajaa professori Antti Niemeä kannustavasta ohjauksesta ja monista neuvoista, joita sain häneltä työni edistyessä.

Työni valvojaa DI Veikko Mäkipäätä ja monia muita IBM:n toimihenkilöitä kiitän kaikista käytännön neuvoista, ja tuesta, jotka edesauttoivat työni valmistumista. Ruotsin IBM:n robottiryhmää haluan kiittää joustavasta yhteistyöstä, ja siitä, että sain käyttööni tarpeelliset mekaaniset työvälineet käytännön kokeilujen tekemiseksi.

Toivon, että työni on hyödyksi Oy Wärtsilä Ab:n Turun telakan robotisointiprojektin edistymiselle ja että työni aikana selvinneitä asiota voitaisiin käyttää hyödyksi varsinaista tuotantosovellusta toteutettaessa.

Helsinki 25.10.1984



Jaakko Perttu

SISÄLTÖ

1.0 Johdanto	1
1.1 Taustaa	1
1.2 Työn tavoitteet	1
1.3 Aiheen rajaus	2
1.4 Työympäristön kuvaus	2
2.0 Aikaisemmat robotiikkatutkimukset	3
2.1 TKK:n robottinäkö tutkimus	3
2.2 Muita tutkimuksia	4
3.0 Kokoonpanotyö	5
3.1 Kokoonpanotyön kulku	5
3.2 Virhetilanteet	7
3.3 Mukautuva ohjaus	8
3.4 Robotin vaikutus tuotesuunnitteluun	10
3.5 Robottiratkaisun valintaan vaikuttavat taloudelliset seikat	11
4.0 Robotti	15
4.1 Kokoonpanorobotin tehtävät	15
4.2 Robottijärjestelmän liitännät ympäristöönsä	16
4.2.1 Käyttäjäliityntä	16
4.2.2 Liityntä tuotannonohjausjärjestelmään	18
4.2.3 Liityntä tietokoneavusteiseen suunnittelujärjestelmään	18
4.2.4 Liityntä ympäröiviin laitteisiin	18
4.3 Kevyeen kokoonpanoon soveltuvia robottityyppejä	19
4.3.1 Antropomorfinen robotti	19
4.3.2 SCARA-robotti	20
4.3.3 Porttaalirakenteinen robotti	21
4.4 IBM 7565:n rakenne ja ohjausjärjestelmä	22
4.4.1 Laitteisto	22
4.4.2 Robotin kinematiikan kuvaus	23
4.4.3 Robotin liikkeiden ohjaus	24
4.4.4 Servo-ohjaus	26
4.4.5 Tarttuja ja työkalut	27
5.0 Sensorit	29
5.1 Käyttöperiaate	29
5.2 Näkö	29
5.3 Voima	31
5.4 Kosketus	31
5.5 IBM 7565:n sensorit	32
5.6 Sensoritiedon prosessointi	33
6.0 Ohjelmointi	34
6.1 Työkierron suunnittelu	34
6.1.1 Törmäysten esto	34
6.1.2 Kiertojen minimointi	35
6.2 On-line-ohjelmointiympäristö	36
6.3 Off-line-ohjelmointiympäristö	36
6.4 Eri ohjelmointimenetelmät	37
6.4.1 Opetus näyttämällä	39

6.4.2	Robottitasoinen ohjelmointi	40
6.4.2.1	AML-ohjelmointikieli	40
6.4.3	Tehtävätasoinen ohjelmointi	42
6.4.3.1	Käsitteet, komponentit	43
6.4.3.2	Ympäristön mallitus	43
6.4.3.3	Prosessin suunnittelu	45
6.4.3.4	Suoritettavan ohjelman luonti	46
6.4.3.5	Simulointi ja visualisointi	47
6.4.3.6	AUTOPASS-järjestelmä	47
6.4.4	Robottien ohjelmointimenetelmien kehityssuuntia	47
7.0	Käynnistinkaapin kokoaminen robotilla	49
7.1	Lähtötilanne	49
7.1.1	Nykyinen tuotanto	49
7.1.2	Materiaalin kulku	50
7.2	Alustava kannattavuuden arviointi	52
7.3	Kokoonpanosovellus IBM RS/1-robotilla	52
7.3.1	Laitteisto	52
7.3.2	Sovellusohjelmisto	54
7.3.3	Kalustusohjelma	55
7.3.3.1	Komponenttien haut	56
7.3.3.2	Komponenttien asennukset	56
7.3.3.3	Komponenttien ruuvaukset	57
7.3.4	Käyttökokemuksia	59
7.3.4.1	Ohjelmankehitys	59
7.3.4.2	Kalustusohjelman toiminta	59
7.4	Järjestelmän kehitystarpeet	60
7.5	Robotisoinnin vaikutus organisaatioon	61
7.5.1	Vaikutus tuotesuunnitteluun	61
7.5.2	Vaikutus työhön	61
7.5.3	Koulutus	62
8.0	Työn arviointi	63
9.0	Kirjallisuutta	64
	Liite A. Sigma-kokoonpanoasema	66
	Liite B. Piirilevyn kokoamisasema	69
	Liite C. Ohjelmalistaukset	71
	Liite D. Käyttäjäliittynän tulosteita	85

KUVALUETTELO

Kuva 1.	Kokoonpanon tilaesitys	6
Kuva 2.	Mukautuva sovitus	9
Kuva 3.	Kokoonpanomenetelmien ominaisuuksien vertailu	12
Kuva 4.	Kokoonpanomenetelmien kustannusvertailu /4/	13
Kuva 5.	Kokoonpanosolu	14
Kuva 6.	Robottijärjestelmän liitynnät ympäristöön	17
Kuva 7.	Antropomorfinen robotti	20
Kuva 8.	SCARA-robotti	21
Kuva 9.	IBM 7565-robottijärjestelmä	22
Kuva 10.	IBM 7565:n kinematiikka	23
Kuva 11.	IBM 7565-robotin ohjausohjelmisto /9/	25
Kuva 12.	Servopiirin kaavio /9/	27
Kuva 13.	Tarttujan sormen rakenne /13/	28
Kuva 14.	Ohjelmankehitys IBM 7535-simulaattorilla /7/	38
Kuva 15.	Suunnittelun integrointi	42
Kuva 16.	Tehtävätasoinen ohjelmointiympäristö /19/	44
Kuva 17.	Nykyiset tuotantomäärät	50
Kuva 18.	Materiaalin kulku kokoonpanoprosessissa	51
Kuva 19.	Komponenttien sijoittelu robotin työalueelle	53
Kuva 20.	Ohjelmiston rakenne	55
Kuva 21.	SKRV-ohjelman yksinkertaistettu tilakaavio	58
Kuva 22.	Sigma-kokoonpanoaseman ohjauslaitteisto	68
Kuva 23.	SEAS-aseman ohjauslaitteisto	70

1.0 JOHDANTO

1.1 TAUSTAA

Ihmistyövoiman käytön kallistuessa samalla kun laitteistojen hinnat halpenevat aiheuttavat yhdessä sen, että yritykset pyrkivät parantamaan kilpailukykyään automaation avulla.

Kappaletavaratuotantoa voidaan automatisoida tuote- ja työvaihekohtaisilla automaattikoneilla, mutta sarjakokojen pienentyessä tarvitaan välineitä, joilla useita työvaiheita voidaan tehdä samalla laitteistolla. Erityisesti kokoonpanotehtävissä tämä on tärkeää.

Ohjelmoitavat teollisuusrobotit suoriutuvat monivaiheisista-kin töistä, mutta kokoonpanossa käsiteltävien kappaleiden usein vaihteleva laatu aiheuttaa helposti ongelmallisia virhetoimintoja. Näiden ongelmien ratkaisemiseksi on tehty laitteiston osalta huomattavaa kehitystyötä, minkä seurauksena nykyisin on saatavilla teollisuustuotantoon soveltuvia kokoonpanorobottilaitteistoja, jotka ottavat huomioon mahdolliset virhetilanteet.

Robottiikkatutkimus keskittyy robotin aisteihin ja niiden liittämiseen robotin ohjausjärjestelmään. Lisäksi tutkitaan erilaisia korkean tason ohjelmointimenetelmiä. Näistä tutkimuksista saadaan soveltamiskelpoista tietoa kokoonpanonem-
netelmistä ja keinoaistien käytöstä.

Teknillisen korkeakoulun säätötekniikan laboratoriossa on tutkittu robotiikkaa yleiskäyttöisen robotin ja prosessitietokoneen avulla ja saavutettu lupaavia tuloksia näköaistin kehittämisessä. Kokoonpanotehtävien toteuttamiseksi tarvitaan vielä lisätyötä lähinnä tuntoaistin liittämiseksi robottiin.

1.2 TYÖN TAVOITTEET

Työn tavoitteena on selvittää kokoonpanotyön robotisoinnissa esiintyviä ongelmia sekä löytää niihin ratkaisuja olemassa olevan robottilaitteiston avulla. Työssä käytetään IBM 7565-robottia, joka kykenee selviämään useista kokoonpanossa esiintyvistä virhetilanteista. Tässä työssä tutkitaan käytännön kokein kyseisen robottityypin soveltuvuutta vahva-virtakytkinkaapin kalustukseen.

1.3 AIHEEN RAJAUS

Työssä keskitytään kevyen kokoonpanon automatisointiin kehittyneellä teollisuusrobotilla korkean tason ohjelmointiympäristössä. Käytännön työssä toteutetaan vahavirtakytkinkaapin kokoonpano robotilla. Vaatimuksena on se, että robotin on toivuttava mahdollisista virhetilanteista.

1.4 TYÖYMPÄRISTÖN KUVAUS

Tähän diplomityöhön liittyvä käytännön soveltamistehtävä toteutettiin Ruotsin IBM:n Järfällan tehtaiden RS/1-robotilaitteistolla. Järfälla:ssa on tutkittu IBM-robottien soveltamista noin kahden vuoden ajan. Tehtaalla on kehitetty yleiskäyttöinen kokoonpanoasema, jonka kalustus voidaan nopeasti vaihtaa. Kyseinen laitteisto soveltuu erityisesti pienien sarjojen tuotantoon.

2.0 AIKAISEMMAT ROBOTIIKKATUTKIMUKSET

Kokoonpanon robotisoinnissa esiintyviä ongelmia on tutkittu useissa eri laboratorioissa. Keskeisiä tutkimuskohteita ovat olleet kappaleiden tunnistus näköaistin avulla, ja kappaleiden sovituksessa tarpeellisen tuntoaistin kehittäminen. Näissä tutkimuksista on saatu seuraavanlaisia tuloksia.

2.1 TKK:N ROBOTTINÄKÖTUTKIMUS

Teknillisen korkeakoulun säätötekniikan laboratorio on tutkinut robotiikkaa ja erityisesti robottinäköjärjestelmiä ja niiden integroimista standardirobotin ohjausjärjestelmään /11/.

Laboratorion laitteistoon kuuluu Asea IRb-6-robotti, johon on liitetty Strömberg 1000-tietokone ohjaamaan näköjärjestelmää. Näköjärjestelmän laitteistoon kuuluu Reticon MC 510A-puolijohdekamera, jonka resoluutio on 50 x 50 kuvapistettä. Kamerasta saatava kuvasignaali siirretään tietokoneen muistiin DMA-kanavan kautta. Robotin ohjausohjelma on suunniteltu Asea IRb-6:a varten, mutta se voidaan helposti saattaa toimimaan muidenkin mikroprosessoriohjattujen robottien yhteydessä. Strömberg 1000 suorittaa koordinaatistomuunnokset kameral ja robotin välillä.

Robotin yhteydessä toimivan näköjärjestelmän käyttö jakautuu kahteen vaiheeseen: opetus- ja tunnistuvaiheeseen. Opetettaessa näytetään järjestelmälle kappaletta, jolloin näköjärjestelmän reunantunnistusalgoritmi havaitsee ja tallentaa muistiinsa opetetun kappaleen. Ääriviivan perusteella lasketaan kappaleelle ominaisia tunnistustietoja, jotka määrittelevät kappaleen, sen sijainnin ja asennon. Uuden kappaleen opettaminen tapahtuu siten, että käyttäjä määrittelee ne ominaisuudet, jotka erottavat uuden kappaleen edellisistä.

Jokaista kappaletta kohden on ohjelmoitu manipulointiohjelma, joka määrittelee tarttumiskohdan ja kappaleen siirron vakioasemaan ja -orientaatioon tarttumisen jälkeen. Ohjausohjelma laskee robotille sellaiset koordinaatit, joilla tarttuminen suoritetaan samasta kohdasta, kuin opetusvaiheessa.

Laitteistoa on sovellettu muovisen palapelin kokoamiseen. Palapelin palat ovat leikattu muovilevystä ja ne on varustettu siten, että robotti kykenee helposti tarttumaan niihin. Robotti havaitsee satunnaisesti orientoituneen kappaleen ääriviivan ja tunnistaa kappaleen ja sen orientaation. Tarttumiskohta kappaleeseen tarkennetaan erillisen asemointitelineen avulla. Tunnistettu kappale kuljetetaan

oikeaan paikkaansa ja pudotetaan paikalleen. Varsinaista sovitustyötä ei tehdä, joten palapelin palojen väliin saat-
taa jäädä rakoja tai ne saattavat osittain mennä päällekin.

Tutkimustyö on suuntautunut keinotekoisen näköaistin tutki-
miseen, eikä erityisesti kokoonpanotyössä esiintyviä sovi-
tus- ja virheestätoipumistehtäviä ole tutkittu, joten
laitteisto ei sovellu varsinaiseen kokoonpanotyöhön.

2.2 MUITA TUTKIMUKSIA

Kokoonpanoon liittyvien ongelmien ratkaisuun on kehitetty
tuotantoonkin soveltuvia laitteistoja. Näistä esimerkkinä
ovat Olivetti:n SIGMA-robotilla toteutettu järjestelmä (ks.
liite A) ja piirilevyjen kokoamiseen tarkoitettu
SEAS-järjestelmä (ks. liite B).

SIGMA-järjestelmä käyttää kappaleiden havaitsemiseen yksin-
kertaista näköjärjestelmää, jossa kappaleen ääri viivoista
otetaan kaksi näytettä halutulta alueelta. Näiden näytteiden
perusteella voidaan päätellä kappaleen oikea asento ym.
Piirilevyjen kokoamiseen suunniteltu SEAS-asema ei käytä
näköjärjestelmää kappaleiden tunnistukseen. Sen sijaan kom-
ponenttien johtimien kohdistus piirilevyn reikiin suorite-
taan erityisesti tätä varten asennetun videokameran avulla.

Kappaleiden sovituksessa käytetään apuna erilaisia voimasen-
soreita. SIGMA-robotin tarttajakourassa on mekaaninen mukau-
tumiselin, johon on liitetty asemasensori valvomaan
mukautumisen suuruutta. SEAS-asema käyttää voimasensoreita
sekä robotin kourassa että kokoonpanoalustassa.

Molemmat järjestelmät soveltuvat kokoonpanotehtäviin. SEAS-
asema tosin vain piirilevyjen kokoamiseen. Järjestelmät ovat
kuitenkin lähinnä laboratoriolaitteita, eikä niitä voi ilman
kehitystyötä käyttää tehdasympäristössä.

3.0 KOKOONPANOTYÖ

Kokoonpanotyössä saatetaan erilaisia kiinteitä kappaleita toistensa yhteyteen siten, että näistä muodostuu haluttu tuote. Osat kiinnitetään toisiinsa esim. ruuvaamalla, erilaisten salpojen, kitkan tai vain painovoiman avulla. Komponentteja käsitellään manuaalisesti, robotin tai kokoonpanoautomaatin avulla.

Kevyessä kokoonpanotyössä käsiteltävien kappaleiden koko on tavallisesti lävistäjällä mitattuna alle 0.5 m ja paino alle 5 kg. Työasemassa käsiteltävien erilaisten komponenttien lukumäärä on tavallisesti 20-30 kpl. Kappaleita käsitellään suurella tarkkuudella. Kappaleet ovat usein lisäksi herkkiä rikkoutumaan. Valmistettavien sarjojen pienen koon vuoksi työtehtävää on usein muutettava.

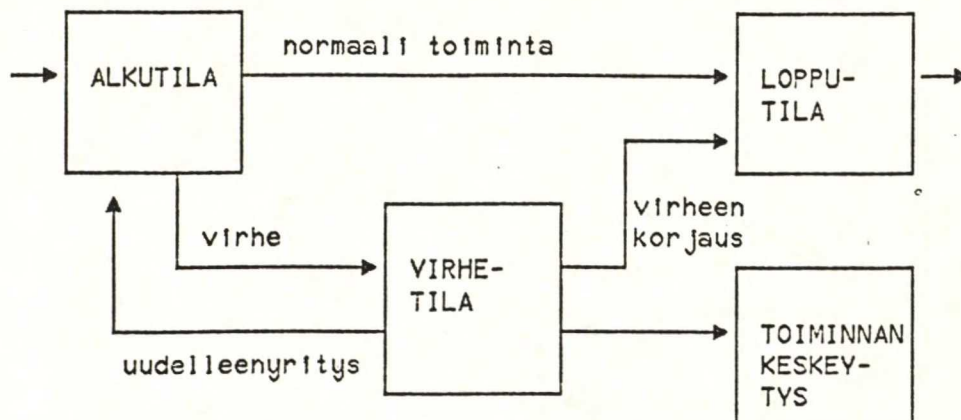
Piensarjatuotannossa tyypillisiä kokoonpanotehtäviä ovat komponenttien asettelu piirilevylle ja mekaanisten osien kokoaminen esim. sähköteollisuudessa. Kokoonpanotehtäviin liittyvät läheisesti myös testaus- ja laaduntarkkailutehtävät.

3.1 KOKOONPANOTYÖN KULKU

Lähtökohtana kokoonpanotyön suunnittelulle on valmis tuote ja siihen tarvittavat komponentit. Tehtävänä on määritellä komponenttien keskinäiset suhteet ja näiden suhteiden muodostamisjärjestys, jolloin lopputuloksena on valmis tuote.

Manuaalisessa kokoonpanossa voi työntekijä päätellä suurimman osan työvaiheista, mutta koneellisessa työskentelyssä, jossa ei ole käytettävissä päättelykykyä on kaikki työvaiheet kuvattava yksiselitteisesti.

Kokoonpanotyön kulku voidaan kuvata sarjana toisiaan seuraavia tiloja /18/. Vähimmäismäärä tiloja muodostuu irrallisten komponenttien ja valmiin tuotteen tiloista. Lisäohjeita antavia välitiloja määritellään tarpeen mukaan. Tilasta toiseen siirtyminen voi tapahtua useaa eri reittiä pitkin siten, että virheestätoipumisrutiinit muodostavat sivuhaaroja virheettömään työn kulkuun (Kuva 1 sivulla 6). Virheenkäsittelyrutiinin tehtävänä on palauttaa kokoonpanotyö aina hallittuun tilaan. Tällä menettelyllä hallitaan kokoonpanon edistyminen häiriöistä huolimatta. Kokoonpano-ohjeen muuttaminen tai uusien työvaiheiden lisäys ei vaikuta muuhun ohjeeseen, joten tilaesitys takaa ohjeen hyvän ylläpidettävyyden.



Kuva 1. Kokoonpanon tilaesitys

Tilaan tulee sisältyä sekä kokoonpanotyön tila että robotin tila. Työn tila sisältää yksittäisten kappaleiden sijainnit ja niiden keskinäiset suhteet. Robotin tilaan sisältyy robotin paikka ja liike, sekä varustus. Tilaan kuuluvia tietoja voidaan täydentää, mikäli havaitaan robotin toiminnassa tai työn kulussa sellaisia virheitä, joita aikaisempi malli ei ottanut huomioon. Mikäli seuraavaa tilaa ei kyetä saavuttamaan, on toiminta keskeytettävä joko kokonaan, tai asianomaisen työkappaleen osalta.

Jotta tämänkaltaiseen toimintaan päästään on työn edistymisen valvonnan oltava mahdollisimman aukotonta. Mahdolliset puutteet valvonnassa paljastuvat koekäyttövaiheessa, jossa ne voidaan vielä korjata täydentämällä robotin ohjausohjelmaa.

Tilaesityksen edut:

- Käyttäjän on helppo ohjelmoida toipumisrutiineja virhetilanteita korjaamaan sitä mukaan, kun virheitä esiintyy.
- Muutosten ja lisäysten teko on helppoa, sillä eri työvaiheet eivät vaikuta välittömästi toisiinsa.
- Oheislaitteiden liittäminen robottijärjestelmään helpottuu.
- Johtaa strukturoituun ongelman ratkaisuun.

Haittapuolena tilaesityksen käytöstä robottisovelluksissa on kasvanut muistitilan tarve, joka ei tosin ole suuri ongelma, sillä kehittyneessä ohjausjärjestelmässä voidaan käyttää suhteellisen halpaa levymuistia laajankin ohjelmiston tallentamiseen.

3.2 VIRHETILANTEET

Kaikkia tapahtumia, jotka saattavat kokoonpanotyön sallitusta lopputilasta poikkeavaan tilaan, pidetään virheinä. Virhetilasta ei voida ilman korjaustoimintoja jatkaa kokoonpanotyötä. Korjaustoimet perustuvat virheiden havaitsemiseen.

Manuaalisessa kokoonpanossa virheet havaitaan ihmisen aistien ja älyn avulla. Kokoonpanoautomaattien käytössä taas pyritään siihen, että virheitä ei esiinny. Robottikokoonpanossa virheisiin on varauduttava varustamalla robotti riittäväillä aisteilla ja toimintaohjeilla, joilla lopputilan oikeellisuus voidaan todeta riittävän tarkasti.

Useimmat virhetilanteet voidaan ennakoida aikaisempien kokemusten perusteella. Käytön aikana ilmenevät virhetilanteet on käsiteltävä aina tapauskohtaisesti.

Työkappaleesta saadaan tietoa näköaistin ja tuntoaistin avulla. Näköaistilla voidaan tarkistaa, että kaikki osat ovat paikoillaan. Robotin tarttujaan yhdistetty tuntoaisti mahdollistaa laadullisien tarkastusten suorituksen. Tunto- tai voimasensoreiden avulla voidaan esim. tarkistaa tietyn osan kiinnityksen pitävyys tai varmistaa, että ruuvi on kiristetty riittävän tiukalle. Robotin kouran avulla tehtävien tarkastusten tulee kuulua läheisesti varsinaisen kokoonpano-ohjelman yhteyteen, sillä näin vältetään ylimääräisten liikkeiden suoritukselta.

Kokoonpanotyö opetetaan robotille siten, että aluksi ohjelmoidaan virheettömän työn kulku. Toiminnan alkuvaiheessa robotti työskentelee valvonnan alaisena. Robotin joutuessa sellaiseen virhetilaan, josta se ei kykene siirtymään osatehtävän lopputilaan, pysähtyy toiminta ja käyttäjä voi ohjelmoida robotille virheenkorjausrutiinin, joka palauttaa kokoonpanon haluttuun tilaan. Miehittämättömässä toiminnassa robotti kykenee näiden toipumisrutiinien avulla itse korjaamaan yleisimmät virhetilanteet ja toiminta tulee vähitellen erittäin luotettavaksi.

Seuraavassa yleisimmät ennakoitavat virhetilanteet, ja niiden korjaustoimet robotilla:

Virhetilanne

Mahdollinen syy ja korjaus

Syöttölaitteesta puuttuu
komponenetti

Osat loppuneet.
Vaihdetaan syöttölaitetta,
tehdään hälytys.

Kourasta puuttuu
komponenetti

Osa pudonnut.
Haetaan syöttölaitteesta uusi osa.

Osa ei mene paikalleen

Osa on väärässä asennossa,
osa on viallinen tai
muuttanut muotoaan.
Siirretään viallinen osa
syrjään ja haetaan uusi osa.
Tehostetaan asennusta
erikoisliikkeillä.

3.3 MUKAUTUVA OHJAUS

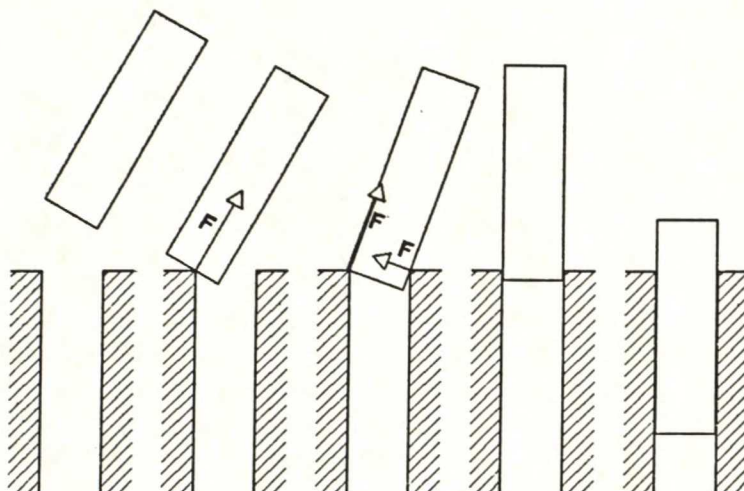
Ympäristön rajoittaessa robotin liikkeitä käytetään rajoituksiin mukautuvaa voimaohjausta. Esimerkkinä rajoitetusta ympäristöstä on tapin liikuttaminen sylinterinmuotoisessa reiässä (Kuva 2 sivulla 9). Ainoastaan tapin kierto ja liikuttaminen sylinterin suunnassa ovat mahdollisia. Kaikkia muita liikesuuntia rajoittavat sylinterin seinämät. Näissä suunnissa ei voida käyttää paikkaohjausta, vaan on käytettävä mukautuvaa voimaohjausta, jolloin sovittaminen hallitaan joustavasti /15/.

Kokoonpanotehtävissä ovat sovitusten tarkkuusvaatimukset usein tiukemmat, kuin robotin toistotarkkuus tai työkappaleen asemointitarkkuus. Lisäksi absoluuttista tarkkuutta heikentävät ympäristöstä tulevat erilaiset häiriöt. Näiden syiden vuoksi mukautuvan ohjauksen käyttö on välttämätön edellytys kokoonpanotehtävän onnistumiselle.

Mukautuva ohjaus voidaan toteuttaa joko passivisesti mekaanisella rakenteella, joka joustaa tiettyihin suuntiin vaikuttavien voimien mukaan, tai aktiivisesti servopiirissä, jolloin sen etuna on ohjelmoitavuus.

Passiivinen mukautuvuus voidaan toteuttaa esim. yhdistämällä paikkaohjattuun robottiin mekaanista jostavuutta vaakasuunnassa rakenteen ollessa hyvin tukeva pystysuunnassa. Myös erilaiset voimiin mukautuvat ranneratkaisut ovat yleisiä. Näiden menetelmien etuna on nopea toiminta, mutta jokaiseen erilaiseen sovellukseen on kehitettävä erilainen mukautumisen elin. Tämä monimutkaistaa rakennetta ja nostaa laitteiston kustannuksia.

Aktiivinen voimaohjaus voidaan toteuttaa servopiiritasolla tai korkeammalla tasolla ns. valvottujen liikkeiden avulla. Valvotulla liikkeellä tarkoitetaan paikkaohjausta, jossa



Kuva 2. Mukautuva sovitin

liikkeen voi pysäyttää jokin ulkopuolinen tapahtuma, esim. tietyn voiman ylitys. Ideaalitapauksessa nämä menetelmät täydentävät toisiaan siten, että valvottuja liikkeitä käytetään lähestyessä liikettä rajoittavaa pintaa. Kappaleen pinnalla liikuttaessa käytetään ohjelmoitavaan voimatakaisinkytkentään perustuvaa ohjausperiaatetta.

Ohjelmoitava voimatakaisinkytkentä voidaan saada aikaan esim. manipuloimalla paikkatakaisinkytkennän 'jäykkyyttä' tai erottamalla toisistaan paikka- ja voimaohjausta soveltavat suunnat.

Paikkatakaisinkytkentää voidaan kuvata seuraavalla yhtälöllä:

$$f = K(p-p_0), \text{ jossa}$$

- f = ohjauksessa käytettävä voima
- p = mitattu työkappaleen todellinen paikka
- p_0 = työkappaleen haluttu paikka
- K = manipuloitava jäykkyyismatriisi

Kummankin menetelmän tehokas käyttö edellyttää tietoa kappaleiden muodoista ja kinemaattisista suhteista. Nämä tiedot tulisi saada geometrisen mallituksen tuloksena. Parhaiten ohjelmoitava mukautuvuus soveltuukin käytettäväksi mallipohjaisten, tehtävätasosten ohjelmointijärjestelmien yhteydessä.

IBM 7565-robotissa on mukautuvuus toteutettu mekaanisesti tarttuvien sormien joustolla. Tämän lisäksi voidaan kouraan asennettuja voima-antureita käyttää ohjaamaan kappaleen sovitusta valvottujen liikkeiden avulla. Tällöin mukautuvuutta ohjataan ohjelmatasolta, mistä seuraa se, että toimintanopeus on pieni (alle 5mm/s).

3.4 ROBOTIN VAIKUTUS TUOTESUUNNITTELUUN

Robotisoinnilla pyritään valmistuksen tuottavuuden parantamiseen, joten robotin käytön tärkein peruste on se, että kokoonpanotyö on mahdollista suorittaa robotin avulla pienemmillä kustannuksilla, kuin muilla menetelmillä.

Robotin käyttö kokoonpanossa edellyttää luonnollisesti sitä, että robotti kykenee suoriutumaan kokoonpanotehtävästä. Tämä seikka edellyttää robotin käytön ottamista huomioon jo valmistettavan tuotteen suunnitteluvaiheessa, joten tuotesuunnittelijoiden tulee olla tietoisia robotin mahdollisuuksista ja rajoituksista. Usein voidaankin jo olemassa oleva tuote saada soveltuvaksi robottikokoonpanoon pienehköillä muutoksilla, esim. standardoimalla komponenttien kiinnitysmenetelmät.

Tuotteen helppo koottavuus on edullista myös kaikille muille kokoonpanomenetelmille kuten manuaaliselle kokoonpanolle tai kiinteälle automaatiolle. Huolellinen suunnittelu takaa myös sen, että mahdollinen tuotteen idean kopiointi ja mekaniikan uudelleen suunnittelu tulee kannattamattomaksi /1/. Mitä aikaisemmassa vaiheessa robottikokoonpano otetaan huomioon, sitä suurempiin säästöihin päästään, sillä aikaisessa vaiheessa tehdyt muutokset kohdistuvat suhteellisen pieneen määrään dokumentteja eikä esim. työkalukustannuksia vielä ole. Muutosten lykkääminen voi lisätä kustannuksia jopa niin, ettei robotisointi enää ole kannattavaa.

Jotta tavoitteisiin päästäisiin on tuotesuunnittelulle asetettava tiettyjä rajoituksia ja suosituksia, joiden avulla otetaan huomioon robotin käyttö kokoonpanossa.

Suosituksia tuotesuunnittelulle /1/:

1. On minimoitava komponenttien lukumäärä. Osien merkityksen ja tarpeellisuuden analysoinnilla voidaan usein vähentää osien lukumäärää n. 40%.
2. On taattava robotin kouran helppo pääsy kohteeseen. Asennussuuntien minimointi vähentää tarvittavia robotin vapausasteita ja mahdollistaa näin yksinkertaisemman ja halvemmän robotin käytön. Suositeltavin asennussuunta on ylhäältä alas jollekin tasomaiselle runkokappaleelle. Esimerkkinä komponenttien asennus piirilevyille.

3. Työkalujen lukumäärä on minimoitava. Edullista olisi, jos kaikkiin kappaleisiin voitaisiin tarttua yleiskäyttöisellä tarttumakouralla. Tämä onnistuu, mikäli jo suunnitteluvaiheessa varataan kappaleeseen tarttumiskohta. Robotin tavallisimmat tarttumat ovat kaksisorminen koura tai alipaineella toimiva imukuppi.
4. On tehtävä tarkkojen komponenttien kohdistus helpoksi käyttämällä erityisiä kohdistustappeja tai käyttämällä viisteitä helpottamaan osien sovittamista. Robotin on kyettävä seuraamaan mahdollisia viisteitä.
5. Automaattisten syöttölaitteiden asettamat rajoitukset on otettava huomioon. Esimerkkinä tärysyöttimissä käytettävien pienikokoisten komponenttien suunnittelu. Komponentteja on voitava käsitellä erilaisten estokappaleiden avulla. Osan orientointi tehtävä helpoksi esim. symmetristen osien käytöllä. Epäsymmetrisiä osia, joiden epäsymmetrisyys kätkeytyy rakenteeseen on vältettävä. Tällaisiin osiin voidaan lisätä erilaisia ohjaustappeja orientoinnin helpottamiseksi. Kappaleet eivät saa olla sellaisia, että ne tarttuvat toisiinsa kiinni syöttölaitteissa pinoutumalla, tai ketjuuntumalla. Vältettäviä muotoja ovat erilaiset kiilat ja avoimet koukkumaiset kappaleet.
6. Kiinnitysmenetelmät tulee standardoida ja tehdä mahdollisimman yksinkertaisiksi. Helpoin kiinnitysmenetelmä robotin kannalta on pikasalvalla toimiva kiinnitys. Menetelmä ei tosin tee osan irroitus-työtä esim. huoltoa varten helpoksi. Mikäli on välttämätöntä käyttää ruuvikiinnitystä, tulee ruuvityypit vakioda ja ruuvien lukumäärä minimoida. Ruuvien lukumäärää voidaan usein vähentää käyttämällä erilaisia tukirakenteita, joissa yhdellä ruuvilla ainoastaan lukitaan kappale paikalleen. Kahta käsivartta vaativat työvaiheet ovat useimmiten mahdottomia yksikätiselle robotille, joten näitä tehtäviä tulee välttää.

3.5 ROBOTTIRATKAISUN VALINTAAN VAIKUTTAVAT TALOUDELLISET SEIKAT

Tuote voidaan koota kolmella eri menetelmällä: manuaalisesti, robotilla, tai kiinteää automaatiota käyttäen. Näistä menetelmistä on valittava kokonaiskustannuksiltaan edullisin. Menetelmän valintaan vaikuttaa oleellisesti tuotannon määrä ja joustavuusvaatimukset. Manuaalinen kokoonpano on joustavin ja kiinteä automaatio jäykin menetelmä (Kuva 3 sivulla 12). Piensarjatuotannossa ei kiinteä automaatio yleensä tule kysymykseen, joten menetelmän valinta on suoritettava manuaaliratkaisun ja robottiratkaisun välillä.

	Manuaalinen	Robotti	Kiinteä automaatio
Paikannustarkkuus	huono	tyyydyttävä	hyvä
Joustavuus	hyvä	hyvä	huono
Monipuolisuus	hyvä	tyyydyttävä	huono
Yleiskäyttöisyys	hyvä	tyyydyttävä	huono
Nopeus	tyyydyttävä	tyyydyttävä	hyvä
Luotettavuus	tyyydyttävä	hyvä	hyvä

Kuva 3. Kokoonpanomenetelmien ominaisuuksien vertailu

Kokoonpanomenetelmien taloudellisten vaikutusten arviointi ja vertailu on vaikeaa, sillä eri menetelmien epäsuorat vaikutukset ovat suuret ja ne voivat vaihdella suuresti kulloisenkin tuotantoympäristön mukaan. Keskeisiä arviointikriteerejä ovat tuotteen läpimenoaika, järjestelmän luotettavuus ja elinaika. Tavanomaiset investointilaskelmat antavat harvoin käyttökelpoista kuvaa jonkin tietyn menetelmän kannattavuudesta, sillä ne eivät ota huomioon suuria epäsuoria vaikutuksia ja järjestelmän joustavuutta. Esimerkiksi kuinka voidaan ottaa huomioon robotin käytön aiheuttaman paremman tuotesuunnittelun tuomat edut? Robotin käyttö edellyttää kokonaisvaltaista ajattelua, jossa kaikkea huomiota ei kiinnitetä lyhyen aikavälin suoriin säästövaikutuksiin. Lisäksi robotin kannattavuuteen vaikuttaa oleellisesti yrityksessä oleva tieto ja taito hyödyntää robottien tarjoamat mahdollisuudet.

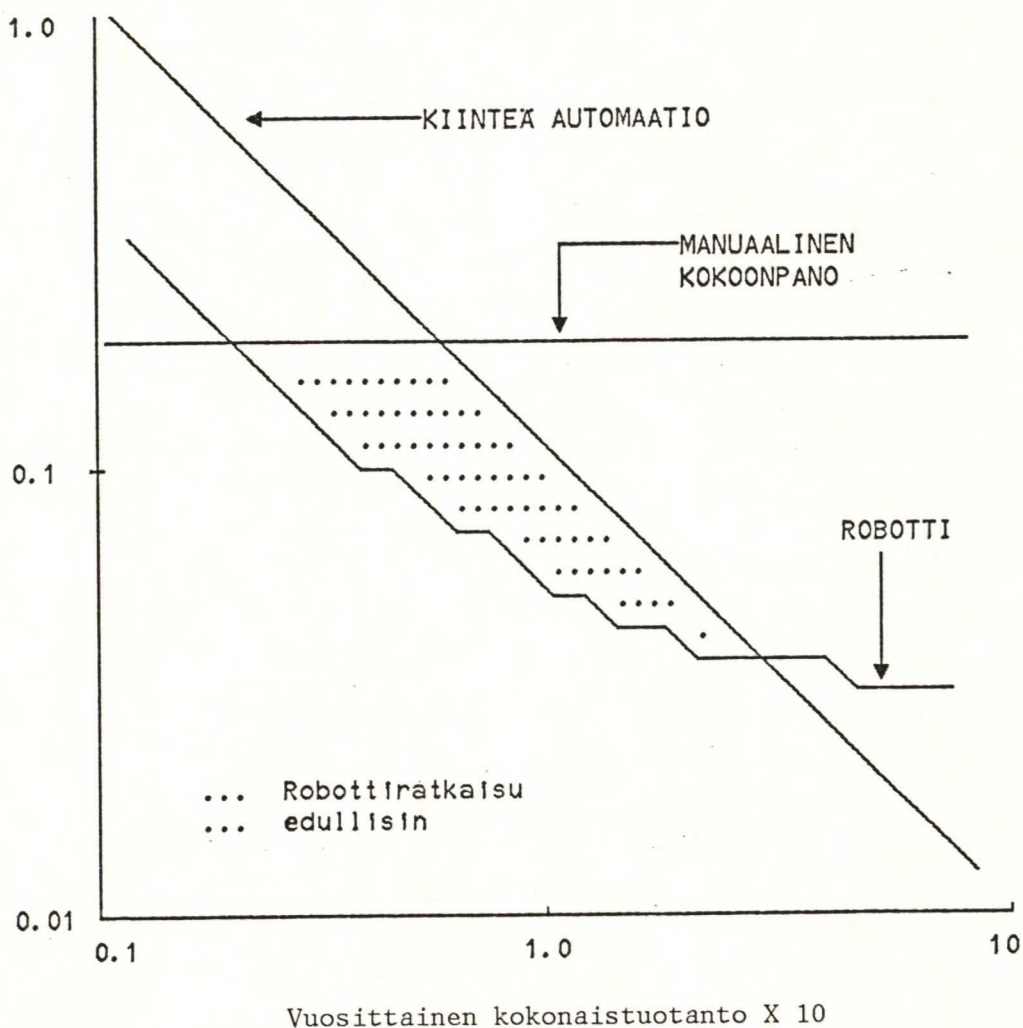
Eri kokoonpanomenetelmien kustannusvaikutuksia on kokemusperäisesti tutkittu, ja joitakin yleisiä suuntaviivoja voidaan antaa. Yleisesti eri kokoonpanomenetelmien kustannusvaikutukset suhtautuvat toisiinsa käyrästä (Kuva 4 sivulla 13) mukaisesti.

Lisäksi on otettava huomioon osien monimutkaisuudesta, työkalukustannuksista, suunnittelumuutoksista ja tuotteiden lukumäärästä ja toimintaohjelman vaihtamisesta aiheutuvat kustannukset.

Tuotantosolun toteutus jakautuu eri osatehtäviin /4/, joiden väliset vuorovaikutukset ovat voimakkaita. Tästä seuraa se, että toteutusprojekti on usein iteratiivinen. Osatehtävät:

1. Tuotesuunnittelu, jossa voidaan käyttää CAD-järjestelmää apuna.

Yksikkökustannus
dollaria/yksikkö



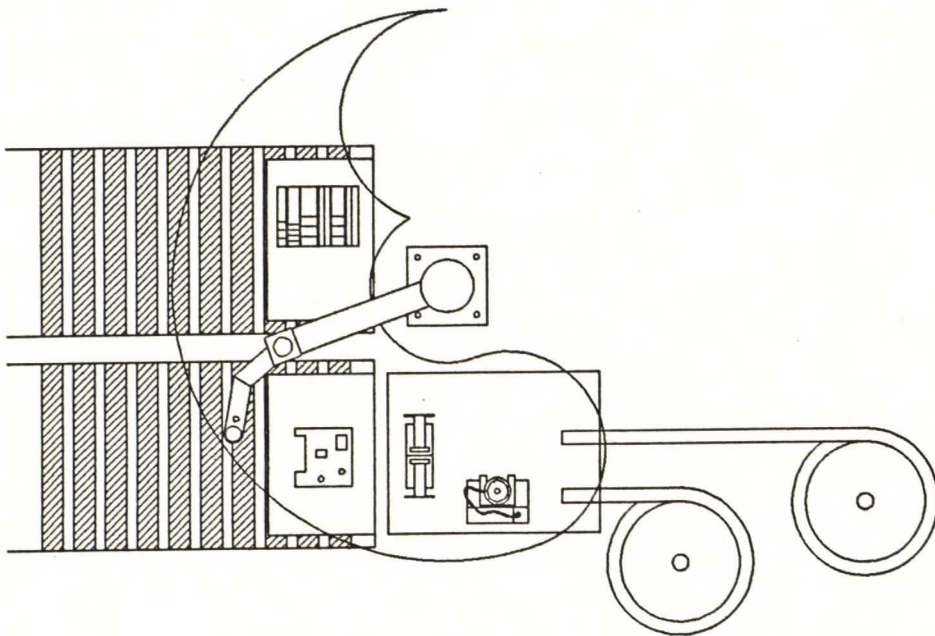
Kuva 4. Kokoonpanomenetelmien kustannusvertailu /4/

2. Kiertoaajan arviointi, mahdollisesti CAD-simuloinnin tai testien avulla.
3. Tuotevalikoiman suunnittelu. Valitaan tuotteet, joiden valmistaminen robotilla on mahdollista ja kannattavaa.
4. Tuotantosolun robotin ja muun laitteiston valinta CAD-simulointien tai testausten avulla.
5. Valmistettavasta tuotevalikoimasta päättäminen.
6. Kustannusanalyysi, jossa varmistutaan projektin kannattavuudesta.

7. Hankintapäätös.

8. Kokoonpanosekvenssin suunnittelu, solun yksityiskohtainen suunnittelu. CAD/CAM-järjestelmän avulla voidaan suunnitella kokoonpanosolun lay-out ja simuloidaan solun toimintaa.

Tuotantosoluun kuuluu robotin lisäksi joukko ympäryslaitteita kuten kuljettimet, syöttölaitteet, robotin käyttämät työkalut ym. (Kuva 5). Mikäli valmistettavien sarjojen koko on pieni, on tuotantosolun kyettävä valmistamaan useaa eri tuotetta. Työn vaihtoon kuluva aika tulee olla mahdollisimman lyhyt, joten robottia ympäröivän kiinteän laitteiston olisi oltava mahdollisimman yleiskäyttöistä.



Kuva 5. Kokoonpanosolu

Tuotantosolun tehokkuutta mitataan sen kapasiteetilla tai yksittäisen tuotteen läpimenoajalla. Kapasiteetin määrää solun hitain komponentti, joka kokoonpanotehtävissä lähes aina on robotti. Kustannuksissa on kuitenkin aina mukana nopea oheislaitteisto. Näiden seikkojen perusteella tuottaa oheislaitteiden hinnan minimointi ja robotin nopeuden maksimointi taloudellisimman tuloksen /4/.

Silloin, kun tuotantosolun ominaisuudet tunnetaan etukäteen, on tätä tietoa käytettävä hyväksi jo tuotesuunnitteluvaiheessa, jotta kokoonpano olisi mahdollista olemassa olevalla laitteistolla.

4.0 ROBOTTI

4.1 KOKOONPANOROBOTIN TEHTÄVÄT

Kokoonpanorobotin keskeisiä tehtäviä ovat kappaleeseen tarttuminen, kappaleen irroitus ympäristöstään, sen vieminen kohteeseensa törmäyksetöntä rataa pitkin, kappaleen sijoitus kokoonpanokohtaan 1. asennus ja viimein kappaleesta irroittautuminen. Lopuksi robotti voi vielä tarkistaa, sujuiko asentaminen virheettömästi.

Näiden tehtävien toteutus vaatii robotilta seuraavia ominaisuuksia:

- Kyky tarttua erilaisista materiaaleista valmistettuihin kappaleisiin, kuten esim. verkkomuuntaja, teräspultti tai muovikotelo käyttäen yleiskäyttöistä tai automaattisesti vaihdettavaa tarttujakouraa siten, että ote on riittävän tukeva eikä kappale vaurioidu.
- Kyky työskennellä nopeasti käyttäen suuria kiihtyvyyksiä.
- Kyky toimia ilman käyttöhenkilökuntaa pitkiäkin aikoja. Tätä varten robotin on kyettävä vaihtamaan työkalujaan automaattisesti ja toipumaan kokoonpanossa sattuvista virhetilanteista.
- Kyky selvittää robotin tai työkappaleen epätarkkuudesta aiheutuneet virheet työkappaleesta saatavan informaation mukaan. Epätarkkuutta aiheuttavat valmistustoleranssit, lämpölaajeneminen, likaantuminen, värinä ym.

Jotta edellämainittuja ominaisuuksia kyettäisiin hyödyntämään joustavasti, on robotti voitava ohjelmoida korkean tason ohjelmointikielellä.

Kokoonpanon ohjelmointi pelkkien paikkakoordinaattien avulla on luonnotonta, sillä se ei ota huomioon kappaleiden valmistustoleransseja /3/. Esimerkiksi kappaleen sovitus jo asennettuun kappaleeseen on tällä menetelmällä virhealtista, sillä varmuutta asennuspaikan tarkkuudesta ei ole. Robotin on kyettävä suorittamaan ns. valvottuja liikkeitä ja mukautumaan kappaleiden epätarkkuuksiin.

Törmäyksetön liikerata suunnitellaan interaktiivisesti ohjelmointi- ja käyttöönottovaiheessa, joten ohjelmaa on voitava muuttaa kesken toiminnan ja jatkaa korjatun ohjelman suoritusta edelleen.

Oheislaitteiden liitäntä robottiin olisi oltava standardoitu, jolloin oheislaitteiden vaihto sujuisi nopeasti ja se olisi mahdollista suorittaa automaattisesti.

Kokoonpanorobottijärjestelmän on ratkaistava kappaleiden siirtoon liittyvät ongelmat. Kappaleitten syöttö robotille voi tapahtua liukuhihnalla, liikkuvien alustojen, tai toisen robotin avulla. Siirtojärjestelmää ei yleensä ohjelmoida jokaiselle tuotteelle erikseen, joten robotin on kyettävä toimimaan yleiskäyttöisten siirtolaitteiden kanssa.

4.2 ROBOTTIJÄRJESTELMÄN LIITYNNÄT YMPÄRISTÖÖNSÄ

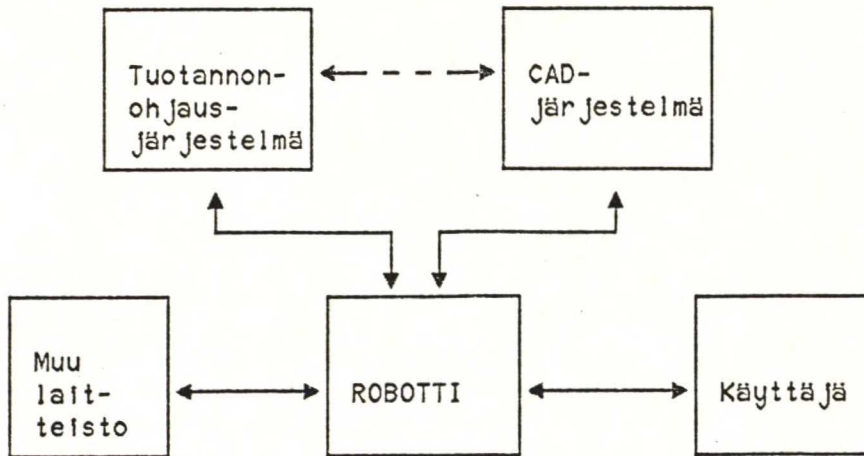
Alati lisääntyvä tuotantojärjestelmän integraatio tuo mukanaan merkittävän kommunikointitarpeen eri järjestelmien välillä. Robotin kyky kommunikoida oheislaitteiden, tuotannonohjauksen, CAD-järjestelmien ja käyttäjän kanssa joustavasti ja luotettavasti mahdollistaa valmistuksen tosiaikaisen ohjauksen ja yhteisten tietokantojen käytön esim. suunnittelun ja valmistuksen välillä (Kuva 6 sivulla 17). Liitännän fyysinen taso muodostuu oheislaitteista kuten pääte ja kirjoitin, liitännästä tiedonvälitysverkkoon, sekä liitännöistä kenttälaitteisiin DI/DO, sarja- tai instrumentointiväylän kautta.

Liitynnän ylemmät tasot voidaan toteuttaa robotin korkean tason ohjelmointikielellä. Tietoliikenneohjelmistot voivat olla käyttöjärjestelmässä valmiina. Kommunikoinnin ja robotin reaaliaikaohjauksen on toimittava toisiaan häiritsemättä.

Tehokkaan ja monipuolisen robotin ohjelmointikielen ominaisuudet tulevat keskeisiksi, kun robottiohjelmaan sisällytetään runsaasti muitakin toimintoja manipulaattorin liikuttelun lisäksi. Kaikki robottijärjestelmään ohjelmoitavat rutiinit tulisi voida ohjelmoida samalla ohjelmointikielellä, jolloin dokumentointi ja ylläpito helpottuisi.

4.2.1 Käyttäjiliityntä

Pääosa toimintaohjeista tulee käyttäjältä ohjelman ja toimintakomentojen muodossa. Robotti raportoi käyttäjälle toiminnastaan: virheistä, suoritusajoista, ym. Turvallisuus on otettava aina huomioon robotin kanssa työskenneltäessä. Tätä tarkoitusta varten on käyttäjällä oltava keinot pysäyttää robotti milloin tahansa riippumatta siitä, mitä robotti on juuri sillä hetkellä tekemässä.



Kuva 6. Robottijärjestelmän liitännät ympäristöön

Kommunikointi on kaksisuuntaista: käyttäjä antaa toiminta-ohjeita robotille ja robotti raportoi tilastaan käyttäjälle. Kommunikoinnin välineinä on tavallisesti tietokonepääte ja kirjoitin sekä mahdollisesti erillinen opetusnäppäimistö ja robotin käyttövoimaan liittyvät ohjauspainikkeet ja merkkivalot.

Käyttäjän syöttämistä ohjeista tärkein on robotin ohjausohjelma. Ohjelma syötetään päätteen ja liikuteltavan opetusnäppäimistön avulla. Painonappien välityksellä ohjataan robotin käyttövoiman kytkeytymistä ym. seikkoja. Turvallisuuden varmistaminen on tärkein peruste painonappien käytölle. Robotti voi aiheuttaa vakaviakin onnettomuuksia, jos turvakytkimet tai muut painonapit eivät olisi selkeitä ja luotettavia käyttää. Robotti raportoi suoritetuista tehtävistä ja niissä tehdyistä virheistä sekä kokoamisajoista.

IBM 7565:n käyttäjäliityntä on toteutettu pääosin alfanumerisen päätteen avulla. Robottia ohjataan joko antamalla yksittäisiä komentoja tai etukäteen kirjoitettuja ohjelmia käyttäen. Mahdolliset virheet voidaan korjata välittömästi, eikä ohjelmaa tarvitse aina virheen sattuessa aloittaa alusta. Ohjelmoinnin helpottamiseksi on olemassa ohjelmisto, jolla ohjelmointi voidaan suorittaa näyttämällä opetusnäppäimistön avulla. Näin pystyvät myös kokemattomat ohjelmoijat tuottamaan toimivia robottiohjelmia. Myös valmiit aliohjelmakirjastot nopeuttavat ja helpottavat ohjelmointia.

Tuotannon ainana IBM 7565 raportoi toiminnastaan kirjoittimen välityksellä. Kirjoitin voi tulostaa jatkuvaa lokilistaa, josta myöhemmin voi seurata kokoonpanotyön kulkua.

4.2.2 Liityntä tuotannonohjausjärjestelmään

Robotin antamia tietoja ovat mm. vuororaportit, virheilmoitukset, tuotekohtaiset valmistustiedot. Tuotannonohjaustietokoneeseen yhdistetty robotti voi raportoida tietoja työvaiheesta, käytetystä kokoonpanoajasta sattuneista virheistä, jotka estävät tuotteen jatkojalostuksen ym. Lisäksi robottijärjestelmä raportoi tuotantohäiriöistä kuten esim. jonkin komponentin loppumisesta.

Tuotannonohjaustietokoneen antamia tietoja ovat yleensä käskyt työn pysäytyksestä, työn vaihdosta, ym. Komentojen perusteella robotti voi esim. kutsua automaattivarastosta tarpeelliset osat ja aloittaa kokoonpano uutta kokoonpano-ohjelmaa käyttäen.

IBM 7565-robotin ohjaustietokone series/1 voidaan sarjalinjaa käyttäen liittää isäntäkoneeseen, joka voi olla esim. tuotantolaitoksen tuotantoa ohjaava tietokone. Robot Control Program huolehtii yhteyden linjakurista ym. Liitännässä siirtyvät tiedot määritellään tuotannonohjaussovelluksen tarpeiden mukaan.

4.2.3 Liityntä tietokoneavusteiseen suunnittelujärjestelmään

CAD/CAM-järjestelmän tietokanta sisältää mm. tietoja kokoonpanossa käytettävien kappaleiden mitoista ja mahdollisesti myös materiaaleista. Käyttämällä näitä valmiita tietoja kasvaa ohjelmoinnin tehokkuus huomattavasti varrattuna siihen, että kappaleiden mitat luodaan uudelleen robotin ohjausohjelmaan.

Mittatietoja voidaan käyttää robottijärjestelmän simulointiin ja ohjelmankehitykseen CAD/CAM-järjestelmällä robotin kinemaattisen mallin avulla off-line-ympäristössä. Ohjelma viritetään kuitenkin valmiiksi aina todellisessa tuotantoympäristössä.

CAD/CAM-järjestelmän tuottamat tiedot olisi voitava siirtää robotin ohjaustietokoneelle, joka kykenee käsittelemään saatua tietoa niin, että tuloksena on valmis tuotantokelpoinen ohjelma. Fyysisen yhteyden lisäksi on sekä robotissa että CAD-järjestelmässä oltava ohjelmisto tiedonsiirtoa varten.

4.2.4 Liityntä ympäröiviin laitteisiin

Robotti ei ole koskaan ainoa tuotteen valmistuksessa käytetty laite, vaan yleensä osa tuotantojärjestelmää, jonka

kanssa robotin on toimittava yhteistyössä. Tästä seuraa tarve kommunikoida muiden laitteiden kanssa. Liityntä voidaan toteuttaa joko yksinkertaisesti DI/DO-linjojen avulla, tai tehokkaammin sarjaväylän (esim. RS 422) tai instrumentointiväylän (esim. IEEE-488) avulla.

Liityntä joustavan valmistusjärjestelmän muihin koneisiin voidaan toteuttaa DI/DO-tasolla. Toiminnan helppokäyttöisyyden takia olisi parempi tehdä rajapinnat koneisiin ohjelmatasolla aliohjelmien avulla. Tällöin liitännän rajapinnan läpi kulkee vain oleellista tietoa, ja mikäli esim. laitteiston toimintaan tulee muutoksia, riittää, kun ko. aliohjelmaan tehdään tarpeelliset muutokset. Esimerkiksi:

PANOSTA(syöttölaite, sensori, toipumisrutiini, yritykset)

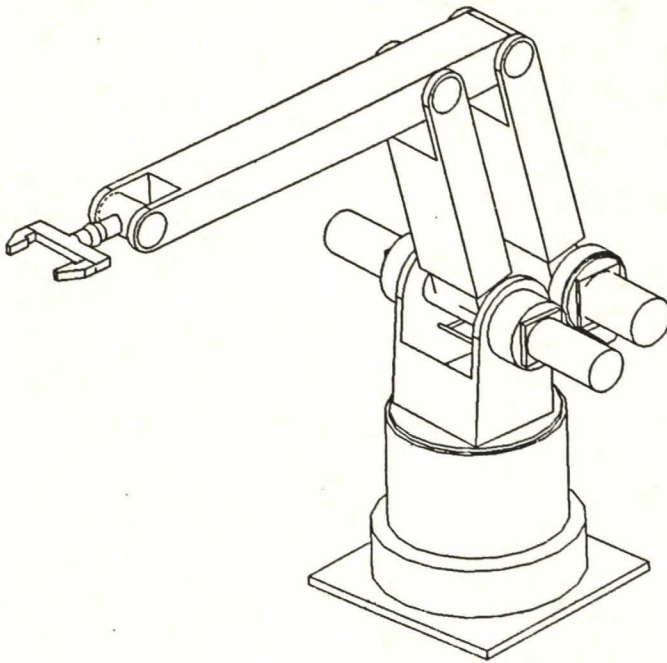
4.3 KEVYEEN KOKOONPANOON SOVELTUVIA ROBOTITYYPEJÄ

Kokoonpanotyöhön, jossa käsitellään pieniä ja keveitä kappaleita soveltuvalta robotilta vaaditaan tarkkuutta ja nopeutta. Robotin kouran on lisäksi kyettävä saavuttamaan mielivaltaisen asento kaikissa työalueen pisteissä. Tämän takia käyttökelpoisessa robotissa olisi oltava mieluiten kuusi vapausastetta.

4.3.1 Antropomorfinen robotti

Robotti koostuu toisiinsa nivelletyistä tukivarsista. Rakenne jäljittelee ihmiskäsivarren rakennetta koostuen olka-, kyynär-, ja rannenivelistä (Kuva 7 sivulla 20).

Robottityypin etuja ovat suuri ulottuvuus ja kompakti rakenne. Ne sopivat hyvin teollisuusympäristöön, jossa esiintyy kuumuutta, pölyä ja tärinää.

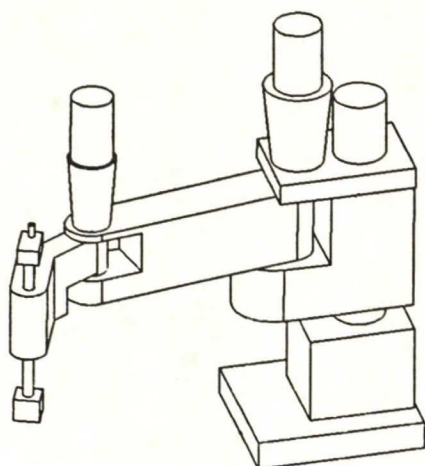


Kuva 7. Antropomorfinen robotti

Mikäli operoidaan suorakulmaisessa koordinaatistossa, kuluu koordinaatistomuunnoksiin suhteellisen paljon laskentakapasiteettia. Robotin dynamiikka muuttuu voimakkaasti käsivarren eri asentojen mukaan, joten mikäli halutaan nopeaa ja tarkkaa toimintaa on tämä otettava huomioon servo-
piirien ohjauksessa. Toistotarkkuus ei ole vakio koko toiminta-alueella, eikä se yleensä ole riittävä vaativiin kokoonpanotehtäviin. Robottityyppejä voidaan käyttää avustamaan kokoonpanosolun muita laitteita. Esimerkkeinä kyseisestä robottityypistä on ASEA IRb-6 ja Cincinnati Milacron T3.

4.3.2 SCARA-robotti

SCARA-robotin (Selective Compliance Assembly Robot Arm) käsivarsi muodostuu kahdesta toisiinsa nivelletystä osasta. Rakenne mahdollistaa vaakasuorat liikkeet X-Y-tasossa. Z-suunnan liikkeet toteutetaan robotin käsivarren päässä sijaitsevan pystysuoraan liikkuvan elimen avulla. Kouran kierto pysty akselinsa ympäri on myös mahdollista, joten vapausasteita on käytettävissä enintään neljä (Kuva 8 sivulla 21).



Kuva 8. SCARA-robotti

Robotin käyttö kokoonpanotyöhön vaatii työkohteelta sitä, että kappaleet voidaan asettaa paikalleen suoraan ylhäältä alas, jolloin neljä vapausastetta riittää. Robotin rakenne mahdollistaa passiivisen mukautumisen kappaleen muotoihin, sillä se on joustava X-Y-suunnassa, mutta jäykkä Z-suunnassa. Robotti kykenee näin sovittamaan itsekohdistuvia komponentteja.

Robottityyppi soveltuu yksinkertaisiin kokoonpanotehtäviin, joissa asennetaan samantyyppisiä komponentteja vaakasuoralle alustalle. SCARA-robotin käyttö voi aiheuttaa tuotteen uudelleensuunnittelua, mutta lopputulos on yleensä edullisempi valmistaa, kuin tuote ennen muutoksia. Esimerkkinä robottityypistä on IBM 7535-robotti.

4.3.3 Porttaalirakenteinen robotti

Robotin rakenne muodostuu suorakulmaisesta kehikosta, jonka sisällä käsivarsi toimii. Robotin käsivarsi liikkuu kehikon rungon johteita pitkin haluttuun paikkaan (Ks. Kuva 10 sivulla 23).

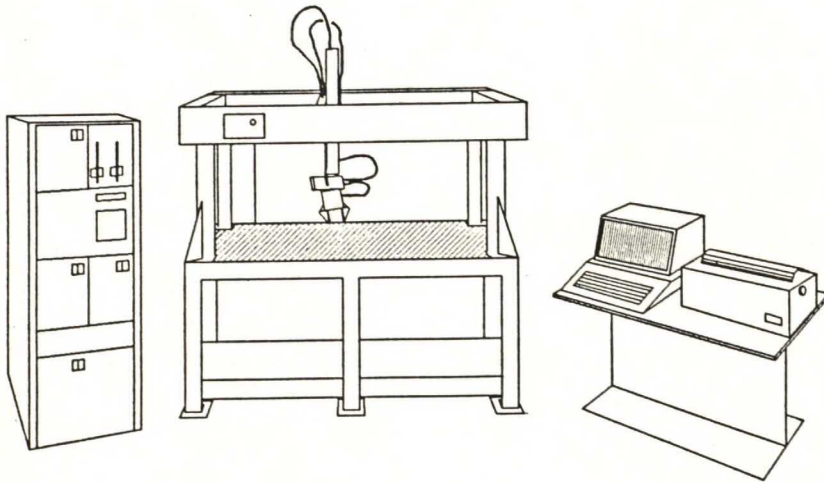
Rakenteen käyttö mahdollistaa sangen yksinkertaisen ohjausjärjestelmän käytön, sillä robotin eri vapausasteiden vaikutukset toisiinsa ovat pienet ja koordinaatisto on jo valmiiksi suorakulmainen. Tukeva rakenne mahdollistaa suurten kiihtyvyyksien käytön sekä takaa hyvän toistotarkkuuden koko toiminta-alueella. Robottityyppi soveltuu vaativaan kokoonpanotyöhön, mikäli koottava kappale vain mahtuu robotin työskentelyalueelle, sillä kehikko rajoittaa kouran

ulottuvuutta. Esimerkkeinä robottityypistä on Olivetti SIGMA ja IBM 7565.

4.4 IBM 7565:N RAKENNE JA OHJAUSJÄRJESTELMÄ

Robottijärjestelmä koostuu manipulaattorista, hydraulijärjestelmästä, ohjaustietokoneesta ja ohjelmistosta (Kuva 9). Porttaalirakenteinen IBM 7565-robotti on suunniteltu tehtäviin, joissa vaaditaan älykästä ohjelmoitavaa manipulaattoria, joka kykenee joustavasti muuttamaan toimintaansa tehtävän tai ympäristön mukaan. Robotti soveltuu kevyeen kokoonpanoon, testaukseen, kappaleiden viimeistelyyn, ja kappaleenkäsittelyyn.

IBM 7565:n vapausasteita on seitsemän. Kuusi liikutteluun ja yksi tarttumakouran avautuman ohjaukseen. Kouran molempiin sormiin kiinteästi asennetut voima-anturit tuntevat kolmeen suuntaan vaikuttavat voimat, joiden aseteltavien kynnsarvojen ylityksiä voidaan valvoa ja suorittaa haluttuja toimia näiden perusteella.



Kuva 9. IBM 7565-robottijärjestelmä

4.4.1 Laitteisto

IBM 7565-robottia ohjaamaan käytetään IBM Series/1-minutietokonetta. Standarditietokoneen käyttö mahdollistaa

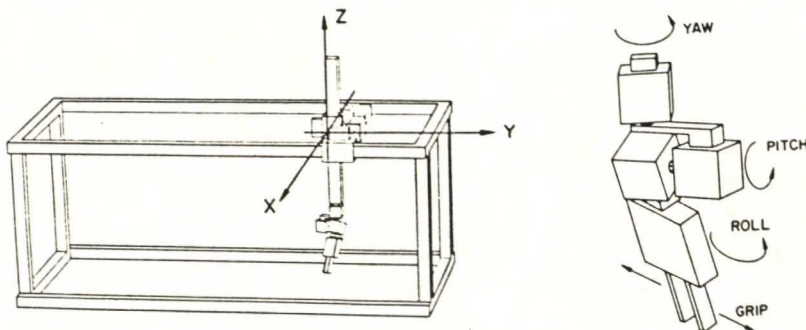
järjestelmän laajennettavuuden ja takaa monipuoliset liitännäismahdollisuudet ympäröiviin laitteistoihin. Tietokoneen laskentatyötä on vähennetty hajauttamalla toimilaitteitten säätöpiirit erillisille analogisille ohjaimille. Series/1:n piirteitä ovat:

- 256 Kb keskusmuisti, josta n. 52 Kb sovellusohjelmille
- liukulukuaritmetiikka
- AI/AO
- DI/DO
- levymuisti 9.3 Mb (esim.)
- EDX käyttöjärjestelmä
- Robot Control Program

Käyttäjää varten järjestelmään on liitetty alfanumeerinen pääte ja kirjoitin. Robotin opettamisen apuna voidaan käyttää erillistä opetusnäppäimistöä. Robotin hydraulikan ohjaamiseksi käytetään kiinteästi robotin runkoon asennettua painonapetta.

4.4.2 Robotin kinematiikan kuvaus

IBM 7565-robotti koostuu suorakulmaisesta kehikosta, jonka sisällä työvarsi toimii (Kuva 10). Kolme lineaarista hydraulimoottoria kuljettavat kouran haluttuun paikkaan X-Y-Z-avaruudessa kehikon sisällä. Robotin käsivarren ran-teessa on kolme kiertoniveltä: kierto pysty- ja vaaka-akselin ympäri sekä kouran pyöritys. Liikkeiden akselit, tai niiden jatkeet leikkaavat toisensa samassa pisteessä. Akselit ovat pareittain kohtisuorassa toisiaan vastaan.



Kuva 10. IBM 7565:n kinematiikka

Rakenteen etuja on tukevuus, joka sallii suuremmat kiihtyvyydet, kuin muut robottirakenteet. Yhden akselin dynamiikka on riippumaton muiden akselien asemasta. Tukevuudesta seuraa myös paikasta riippumaton suuri tarkkuus. Robotin

rakennetta voi myös helposti muuttaa pidentämällä tai lyhentämällä sen suoria osia /9/. Esimerkkirobotissa on kuusi vapausastetta, mutta kokoonpanon niin salliessa voidaan valita myös kolme tai neljä vapausastetta, sillä ylimääräiset nivelet heikentävät aina tarkkuutta.

4.4.3 Robotin liikkeiden ohjaus

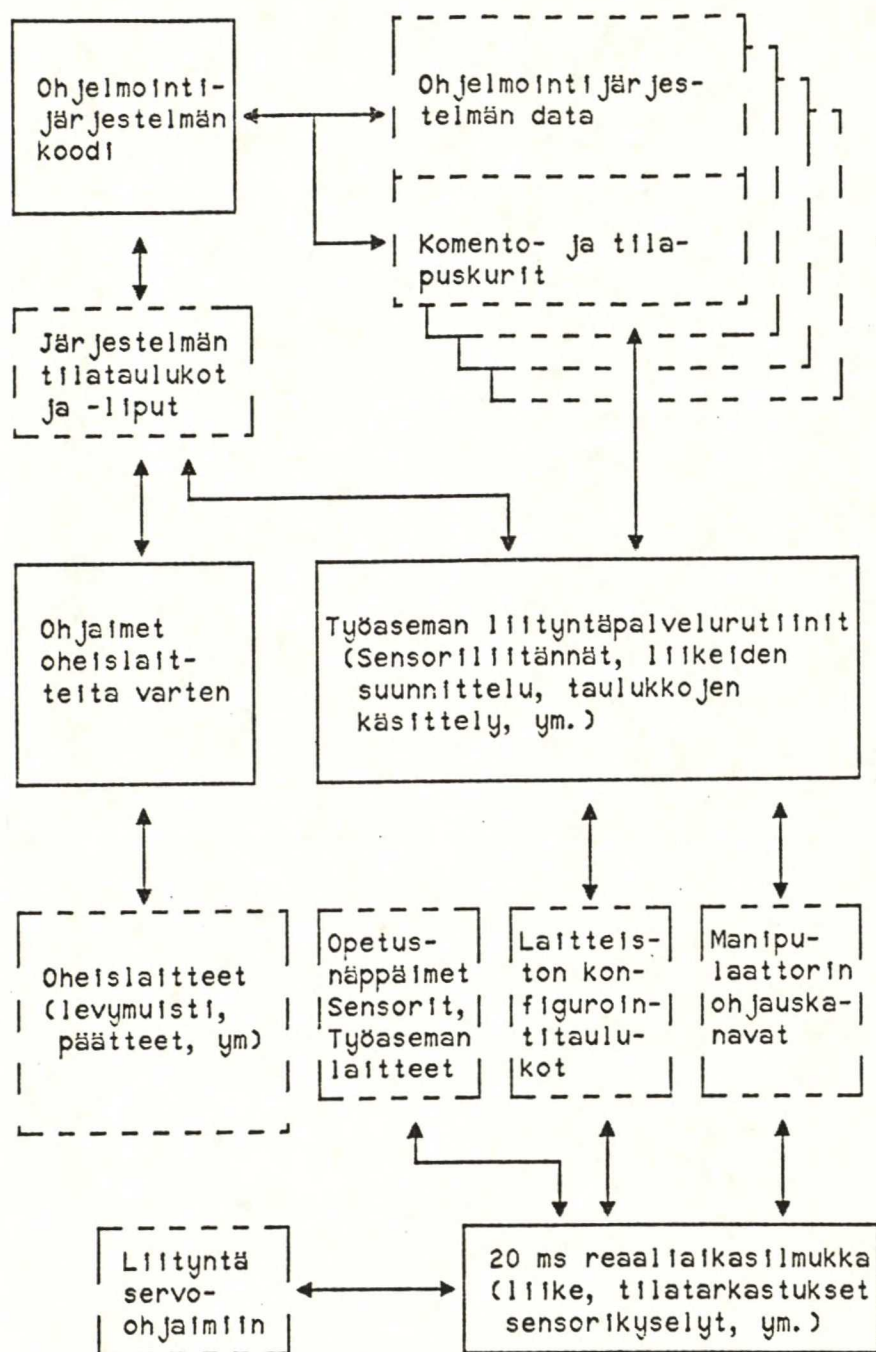
Series/1:n EDX-käyttöjärjestelmän alaisuudessa toimii robotin ohjaamiseen tarvittava vakio-ohjelmisto RCP (Robot Control Program), jonka rakenne on kaavakuvan (Kuva 11 sivulla 25) mukainen. Ohjausjärjestelmän taulukoiden avulla voidaan laitteisto konfiguroida tarpeen mukaan.

20 ms:n reaaliaikasilmukka suorittaa kaikki robotin toimintaan liittyvät toimet kiertokyselyperiaatteella. Ohjelma käynnistyy välittömästi series/1:n käynnistämisen jälkeen. Kunkin kierroksen lopuksi ohjelma nollaa valvontasignaalin. Mikäli nollaus ei tapahdu 25 ms:n kuluttua signaalin asettamisesta, on kyseessä joko ylikuormitus tai muu virhetilanne. Tällöin RCP kytkee robotin hydrauliiikan pois päältä ja toiminta keskeytyy.

Robotin hydraulimoottoreita ohjataan analogisilla servosäätäjillä, jotka saavat uudet asetusarvot 20 ms välein. Sensoriohjaus ei sisälly servopiiriin, vaan ohjaukset sensoritietojen perusteella toteutetaan ohjelmatasolla. Näin sensoreita on joustavampi käyttää, mutta toiminta on suoraa ohjausta hitaampaa. Sensoriohjausta käytetään tosin kappaleen välittömässä läheisyydessä, jossa liikkeiden tuleekin olla hitaita.

Laitteiston konfigurointitaulukossa on kuhunkin moottoriin liittyvät tiedot, kuten liikkeen maksimi- ja minimiasema, suurin sallittu kiihtyvyys ja hidastuvuus, suurin sallittu nopeus ja suurin sallittu poikkeama asemasta. Näistä tiedoista yhdessä liikekäselyn kanssa RCP laskee kunkin moottorin ajo-ohjelman siten, että liikkeet alkavat ja päättyvät yhtäaikaan. Siirtymät x,y,z-koordinaatistossa ovat näin suoriinvaaisia.

Servopiirit ovat erilliset, josta seuraa se, että käyttöjärjestelmän kannalta kaikki robotin vapausasteet ovat samanarvoisia. Tämän ansiosta ohjausjärjestelmällä voidaan hallita monentyyppisiä manipulaattorikonfiguraatioita. Oletuksena on, että manipulaattorin liikevarsien välillä ei ole dynaamista riippuvuutta. Tällainen oletus voidaan tehdä käytettäessä ortogonaalista rakennetta /9/. Ohjaustietokoneeseen on mahdollista kytkeä enintään 14 servo-ohjainta, joilla on mahdollista ohjata joko kahta täydellistä robottikäsivartta tai useampia yksinkertaisia.



Kuva 11. IBM 7565-robotin ohjausohjelmisto /9/

Käsivarren liikenopeutta ohjataan siten, että sitä kiihdytetään vakiokiihtyvyydellä, kunnes saavutetaan haluttu nopeus. Liikkeen pysäytystä edeltää vastaavasti vakiohidastuvuus. Kiihtyvyyden ja hidastuvuuden maksimi-arvot on tallennettu asianomaisiin taulukoihin.

Manipulaattorin paikkaa valvotaan kerran 20 ms:ssa. Jos poikkeama on liian suuri, katkaisee ohjausjärjestelmä tehonsyötön robotille, ja robotti pysähtyy. Tässä oletetaan, että pysähtyminen on turvallista milloin tahansa ja missä tahansa. Mikäli robotti on yhteistoiminnassa muiden laitteiden kanssa, ei pysähtyminen aina ole turvallisin vaihtoehto. IBM 7565-järjestelmä on tältä kannalta itsenäinen yksikkö, jolloin näitä vaaroja ei ole.

4.4.4 Servo-ohjaus

Kutakin robotin moottoria ohjaa analoginen PD-säädin, jonka toimintaa ohjaa ja valvoo RCP (Robot Control Program). Säätimet on sijoitettu kukin omalle piirikortilleen series/1-tietokoneen yhteyteen (Kuva 12 sivulla 27).

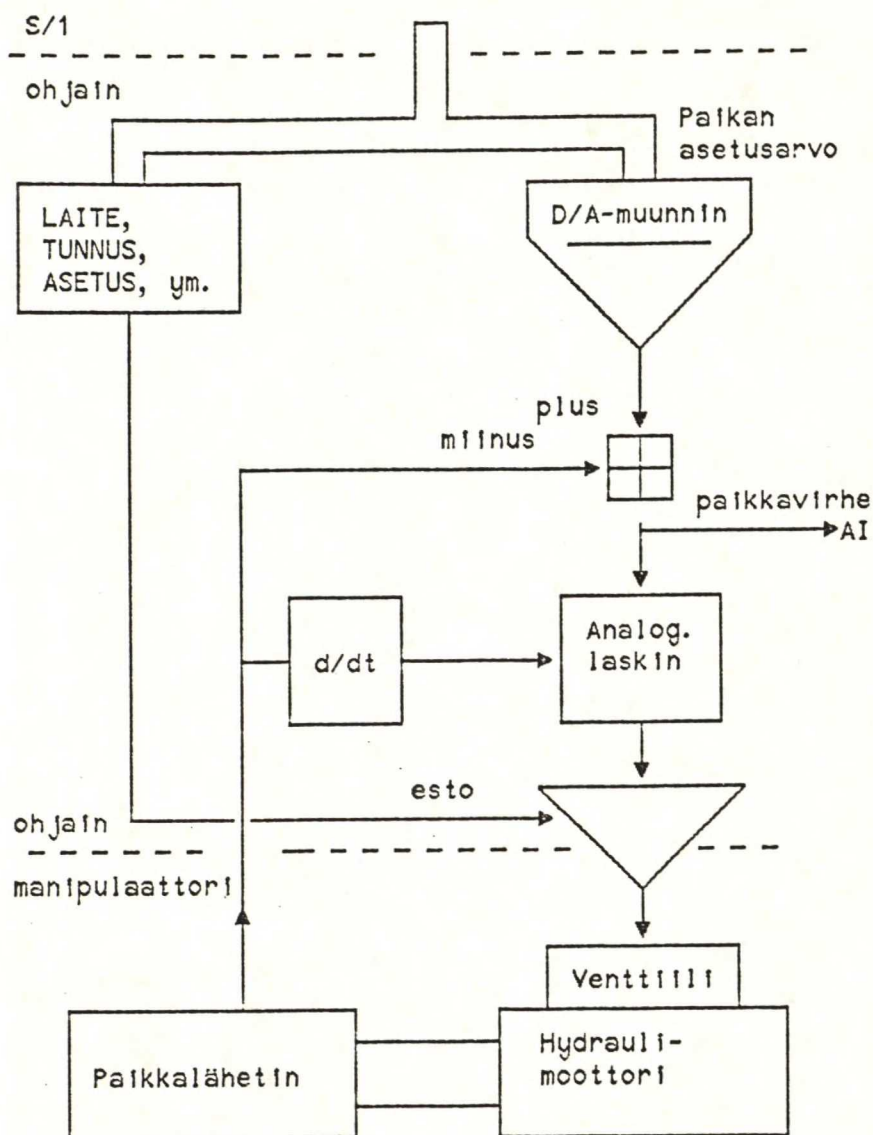
Servon ohjauskortti ja Series/1 välittävät toisilleen seuraavia tietoja:

1. Paikan asetusarvon asetus 16 bit:n kokonaislukuna kahden komplementtiesityksenä.
2. Paikan virheen luku. Virhe luetaan samoin 16 bit:n kokonaislukuna. Virhesuuretta käytetään uusien asetusarvojen laskentaan sekä pysäytyksessä ilmoittamaan, kun haluttu paikka on saavutettu. Mikäli virhe ylittää sallitun rajan, pysähtyy robotin toiminta automaattisesti.
3. Laitteen tunnisteiden luku ja toiminnan valvonta. 16 bit:n koodi identifioi liitetyn laitteen tyyppin ja konfiguraation. Koodi tarkastetaan ohjelmiston initialisoinnin yhteydessä. Koodia käytetään myös diagnostisiin tarkoituksiin.

Suoria robottivarsia liikutetaan lineaarihydraulimoottoreilla. Moottorit liikkuvat tukivarsien mukana, joten teho-painosuhteeltaan edullisten hydraulimoottorien käytöllä saavutetaan hyvät dynaamiset ominaisuudet. Lineaarimoottorit muodostuvat mäntäohjatuista rullista, jotka tahdistettuina painavat aaltomaista kiskoa saaden lineaarisen liikkeen aikaan. Takaisinkytkentä saadaan magnetostriktiiviseltä paikka-anturilta.

Tarttumakouran pyörivät nivelet saavat voimansa hydraulisilta siipimoottoreilta. Takaisinkytkentä saadaan potentiometreiltä.

Servopiirit viritetään robottijärjestelmään kuuluvan diagnostiikkaohjelmiston antamien ohjeiden mukaan. Jokainen servopiiri viritetään erikseen. Ensin viritetään käsivarsi ilman kuormaa. Tämän jälkeen robotille annetaan 2.5 kg:n virituspaino, joka vastaa robotin maksimikuormaa. Säätopiirit viritetään kuorman kanssa. Diagnostiikkaohjel-



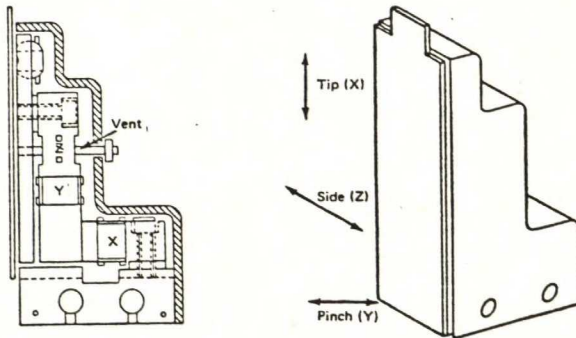
Kuva 12. Servopiirin kaavio /9/

miston ansiosta säätöpiirien viritys ei vaadi käyttäjältä välttämättä servopiirien toiminnan tuntemista.

4.4.5 Tarttuja ja työkalut

Tarttuja muodostuu kahdesta suoraviivaisesti liikkuvasta, yhdensuuntaisesta 'sormesta' (Kuva 13 sivulla 28). Paikkatakaisinkytkentä saadaan potentiometrilta. Tarttujan sormiin on rakennettu mekanismi, joka sallii n. 0.1 mm:n

mukautuvan liikkeen kappaleeseen vaikuttavien voimien suuntaan. Sormien välille on asennettu infrapuna-LED-kennot, joiden avulla voidaan havaita sormien välissä oleva kappale. Sormiin asennetut voima-anturit tutkivat sormissa vaikuttavia voimia kolmeen suuntaan.



Kuva 13. Tarttujan sormen rakenne /13/

Tuntoaistilla varustettu tarttuja on melko yleiskäyttöinen. Käyttäjä voi modifioida tarttumispintaa esim. asentamalla tarttumista helpottavat kynnet sormien päihin. Käyttäjä voi myös korvata standarditarttujan jollakin erikoistyökalulla.

Yleiskäyttöinen tarttuja suoriutuu useimmista kokoonpanossa esiintyvistä tehtävistä vaihdettavien työkalujen avulla. Tärkeimmät aputyökalut ovat erilaiset pihdit ja moottoroidut ruuvimeisselit.

Robotti tuntee kourassa olevaan työkaluun kohdistuvat voimat kouran voimasensoreilla. Työkalujen käyttö tuo mukanaan lisää mahdollisia virhetilanteita, joista robotin on selvittävä. Tämän takia ulkoiset työkalut on varustettava omilla sensoreilla, joiden on kytkeydyttävä ohjausjärjestelmään automaattisesti. Ulkopuolisten työkalujen sensorit kytketään suoraan series/1:n AI- tai DI-korttiin. Esimerkkinä pneumaattisen ruuvimeisselin ruuvi-imun alipaineanturi, jolla havaitaan ruuvin asennon virheet tai ruuvin puuttuminen kokonaan. Työkalut sijaitsevat liikuteltavalla alustalla, paletilla, joka telakoidaan robotin työalueelle. Telakoinnin yhteydessä kytkeytyvät käyttövoima sekä erityyppiset sensorit.

5.0 SENSORIT

5.1 KÄYTTÖPERIAATE

Sensorit ovat laitteita, joilla robotti saa tietoa itsestään ja ympäristöstään toimintansa aikana. Itsestään robotti tarvitsee tietoja tarkan aseman saavuttamiseksi sekä itse-diagnostiikan tarpeisiin. Kokoonpanotehtävissä tulee työkappaleesta saatavan tiedon merkitys keskeiseksi, sillä näissä monimutkaisissa työkiertoissa sattuu usein odottamattomia tilanteita, joista robotin on saatava tieto, jotta asianmukaiset korjaustoimet voidaan käynnistää /3/.

Tyypillisiä ulkoisten sensoreiden käyttötapoja ovat:

1. Toimintavaihtoehtojen määrittäminen
2. Laadunvalvonta, toiminnan tarkastus
3. Kappaleen tutkiminen
4. Mukautuminen ulkoisiin häiriöihin
5. Liikkuvan kohteen seuraaminen

Robotin toiminta määräytyy tavallisesti ohjausjärjestelmässä olevan mallin perusteella, ei siis todellisen ympäristön. Malli voi sijaita ohjelmakoodissa, mutta parempi ratkaisu on käyttää ohjelmakoodista riippumatonta sisäistä mallia. Sensoreita käytetään sisäisen mallin ja todellisen ympäristön välisten ristiriitojen havaitsemiseen. Täysin mahdollista on myös päivittää sisäistä mallia sensoreista johdetun tiedon perusteella. Esimerkiksi syöttölaitteiden lämpölaajenemisesta aiheutuva siirtymä voidaan ottaa huomioon robotin liikesarjassa. Ongelmana on satunnaisen epämääräisyyden ja työympäristön pysyvän muutoksen erottaminen toisistaan. Mikäli virheet ylittävät sallitut rajat, ei mallin ylläpito onnistu, ja toiminta on keskeytettävä.

5.2 NÄKÖ

Robotin näköaistilla tarkoitetaan optisin keinoin saatua tietoa toimintaympäristöstä. Antureina käytetään videokameroita tai yksittäisiä valokennoja. Videokameralla saatava kuva esitetään matriisina, jonka alkiot ovat erisävyisiä kuvapisteitä. Sävyarvoja voi olla kaksi tai enemmän. Mat-

riisikuvaa voidaan käsitellä eritasoisen informaation saamiseksi. Seuraavassa esimerkkejä eritasoisista tiedoista:

1. Kappaleen 2-D ääriviivaprojektio. Kappale tunnustetaan, lasketaan sen asema ja orientaatio. Laskenta voidaan suorittaa nopeasti. Käytetään kaksisävyistä kuvaa.
2. 3-D kappaleen 2-D perspektiivikuvantojen käyttö. Voidaan tunnistaa yksityiskohtien asentoja. Vaatii pitkän laskenta-ajan, minkä takia se ei vielä sovellu reaaliaikakäyttöön.
3. Kahden kuvan yhdistäminen kolmiulotteiseksi malliksi.
4. Kohteen eri yksityiskohtien tunnistaminen ja yhdistäminen eri näkökulmista otetuista kuvista.

Näköjärjestelmiä käytetään nykyisin kappaleen tunnistukseen kiinteästi sijoitetun kameran avulla. Kappale täytyy ennakolta näyttää järjestelmälle, joka sitten tunnistaa samamuotoiset kappaleet ja laskee kohteen paikan ja orientaation.

IBM on kehittänyt kokeilunäköjärjestelmän, joka on tarkoitettu toimimaan IBM 7565-robottilaitteiston yhteydessä /8/. Näköjärjestelmä ohjelmoidaan AML-kielen lisäyksen AML/V avulla. AML/V sisältää kuvainformaation käsittelyyn tarvittavat aliohjelmat, joita AML-ohjelma voi kutsua. AML/V käsittää ohjelmat kameran ohjausta, kuvan hahmottamista ja varsinaista kuvankäsittelyä varten. Näiden perustoimintojen avulla voi käyttäjä ohjelmoida korkeamman tason funktiota kuten esim. hahmontunnistus.

Tietojenkäsittelylaitteistojen tehon ja muistikapasiteetin kasvaessa tulee ensisijaisena kehittämiskohteena olemaan harmaasävykuvien käsittely, mikä pienentää kohteen valaistukselle asetettavia vaatimuksia ja parantaa tunnistuksen luotettavuutta.

Nykyisten näköjärjestelmien liitântä robotin ohjaukseen toteutetaan ohjelmointikielen tasolla siten, että ohjelmassa voidaan testata kappaleen tunnistusta ja asentoa. Liitântä on siis mahdollisimman yksinkertainen. Näköjärjestelmän tehokas käyttö kokoonpanotehtävissä edellyttää nykyistä korkeamman tason tietojen käsittelykykyä.

3-D kappaleen tunnistaminen voi tapahtua siten, että lasketaan kappaleen stabiilit asennot ja verrataan kameralta saatua kuvaa tähän. Laskentaa voidaan edelleen kehittää siten, että projektion laskenta voidaan suorittaa mielivaltaisesta suunnasta ja että myös kulmat ja muut rajapinnat tunnistetaan ja verrataan saatua informaatiota sisäiseen kolmiulotteiseen malliin.

5.3 VOIMA

Voima-anturit antavat kappaleiden sovituksessa välttämätöntä tietoa kappaleiden välillä vallitsevista voimista. Mikäli robotin ranteeseen vaikuttavat voimat ja vääntömomentit mitataan, voidaan saada aikaan kehittynyt mukautumiskyky, joka avulla osien väliset tiukatkin sovitteet voidaan hallita. Myös mukautuminen liikkuviin kohteisiin tulisi näin mahdolliseksi.

Monidimensioisilla venymäliuskamittasilloilla mitataan eri suuntiin vaikuttavat voimat ja vääntömomentit. Niiden käyttö edellyttää tosin suurehkojen (6 x 8) matriisien käsittelyä reaaliajassa, mikäli voimatakaisinkytkentä halutaan sisällyttää manipulaattorin säätöpiiriin /3/. Sormien välillä vallitsevien voimien tunteminen mahdollistaa varman ja tukevan tartunnan erilaisiin kappaleisiin. Sormien voima-antureita voidaan rajoitetusti käyttää myös komponenttien sovituksessa.

5.4 KOSKETUS

Kosketusanturin tehtävänä on tutkia kappaleen olemassaoloa ja mahdollisesti asemaa suhteessa tarttumakouraan. Ideaalinen kosketus- tai tuntoanturi kykenisi samankaltaiseen toimintaan, kuin ihmisen iho eli tunnistamaan kosketeltavan esineen ja tutkimaan sen pinnan materiaalia. Teollisuuskäyttöön soveltuvan, mielivaltaisen muotoisen esineen tunnistavan kosketusmatriisin tulee olla erottelukyvyltään n.3mm ja herkkyydeltään 10g /17/.

Tuntoelin muodostuu siis tuntomatriisista, jonka jokainen alkio on erikseen luettavissa. Jotta johdotuksesta ei tulisi epäkäytännöllisen paksua, on tiedonsiirto tuntoelimen ja robotin ohjaimen välillä toteutettava sarjamuotoisena. Tätä varten tarvitaan pyyhkäisylogiikkaa, joka tulee olla integroituna tuntoelimen välittömään läheisyyteen. Tuntomatriisin alkiot voidaan lukea joko yhden tai useamman bitin tarkkuudella samoin, kuin kuvankäsittelyjärjestelmissä.

Tuntoelimen prosessointi voi tapahtua täysin analogisesti kuvankäsittelysovellusten kanssa eli tunnistus tapahtuisi vertailemalla kosketuskuviota ennalta annettuun mallikuviin. Nykyisin vielä ratkaisemattomia ongelmina ovat tehonkulutus, lämpeneminen sekä riittävän kestävä ja samalla herkän tuntomateriaalin kehittäminen.

Nykysin käytössä olevia kosketuselimiä ovat erilaiset rajakytkimet, jotka toimivat joko mekaanisen kosketuksen perusteella tai valvoen yli- tai alipaineen muutoksia tarttumaelimen ja käsiteltävän kappaleen välillä. Nämä

ilmaisevat ainoastaan kappaleen olemassaolon tai puuttumisen.

5.5 IBM 7565:N SENSORIT

Robotilla on vakiokokoonpanossaan kuusi voima-anturia, jotka ovat sijoitettu robotin tarttujan sormiin, kolme kumpaankin. Niiden avulla robotti mittaa kouraan vaikuttavat voimat, mutta ei siihen kohdistuvia vääntömomentteja. Sensorit mahdollistavat komponenttien sovituksen valvottuja liikkeitä käyttäen. Sensoreilla voidaan myös tutkia kappaleiden työn kulkuun vaikuttavia ominaisuuksia.

Voimasensorit on nimetty AML-ohjelmointikielessä seuraavasti:

Pystysuorat voimat	(SLT,SRT)
Puristusvoimat	(SLP,SRP)
Sivuttaisvoimat	(SLS,SRS)

MONITOR-komennon avulla voidaan asettaa halutut ehdot, jotka liipaisevat halutun toiminnan, esim. pysäyttää liikkeen. MONITOR-komennossa määritellään sensorit, testin tyyppi, valvontarajat, valvontataajuus sekä mahdollinen aliohjelma, jota kutsutaan liipaisun tapahduttua. Valvontataajuus ja aliohjelmakutsu annetaan vain tarvittaessa.

MONITOR(<sensorit>,<testin tyyppi>,<raja1>,<raja2>,
<taajuus>,<aliohjelmakutsu>)

Esimerkkinä liikekomennosta, jossa määritellään ensin robotin liikuteltavat moottorit (part1) ja moottoreiden uudet asema-arvot(part2). Liike pysähtyy monitor-ehdon täyttyessä.

MOVE(Part1,Part2,monitor)

IBM 7565:n kosketusanturina toimii infrapuna-LED-kennot tarttumakouran sormien välissä. AML-ohjelmointikielen kanalta rajakytkin on normaali digitaalitulo, joka on erikseen nimetty. Sensoreiden arvoja luetaan AML-kielen SENSIO-käskyllä, esim:

SENSIO(LED,0)

Sensoreita käytetään mm. robotin kalibroimiseen työkappaleen suhteen. Näin eliminoidaan eri kohteiden siirtymisestä aiheutuvat virheet. Mikäli virheitä kuitenkin tapahtuu, voidaan ne sensoreiden avulla havaita ja korjata. Sovituksessa käytetään normaalisti mekaanista mukautuvuutta, mutta voimasensoreiden käytöllä voidaan valvoa sovituksen onnistumista, ja tarvittaessa suorittaa avustavia liikkeitä.

5.6 SENSORITIEDON PROSESSOINTI

Suoraan sensorilta saatava informaatio on harvoin suoraan käyttökelpoista, vaan sitä on käsiteltävä, jotta se saataisiin robotin ohjausohjelman ymmärtämään muotoon. Prosessin määrä on riippuvainen kulloinkin tarvittavan tiedon tasosta. Mitä pitemmälle jalostettavasta tiedosta on kyse, sitä kauemmin laskenta saa kestää, sillä tiedon perusteella tehtävät päätökset vaikuttavat kaukaisempaan tulevaisuuteen. Vastaavasti alhaisella tasolla oleva prosessointi on tapahduttava reaaliajassa. Nämä seikat puolustavat sensorikäsittelyn jakamista hierarkisiin tasoihin, joilla kullakin tietojenkäsittelytarve on kutakuinkin sama /6/.

Hierarkiatasot ja tehtävät voidaan jakaa seuraavasti /6/:

0-taso: Kuvainformaatio kuva-alkiotasolla, väri, etäisyys. Toiminta on synkronissa prosessin kanssa. Tällä tasolla ei vertailla sensoreiden välisiä mahdollisesti ristiriitaisia tietoja. Ei käytetä erillistä ympäristömallia.

1-taso: Lasketaan vektoreita, nopeuksia, kulmia. Asynkroninen toiminta, ei verrata malliin.

2-taso: Tunnistetaan kappaleita, verrataan saatua tulosta ympäristömalliin. Mallia voidaan tutkia tunnetuista katselukulmista (2-D).

3-taso: Malli sisältää ympäristön kokonaisuudessaan. Mallituksessa käytetään kolmedimensioista tilavuusmallia, jossa kappaleiden väliset suhteet on kuvattu.

Eri hierarkiatasot kommunikoivat keskenään siten, että kullakin rajapinnalla operoidaan suureilla, joita molemmat tasot kykenevät käsittelemään. Prosessimallin ja sensoripalautteen antamaa informaatiota vertaillaan kaikilla tasoilla ja tarvittaessa päivitetään mallia tai annetaan virheilmoitus ylemmälle hierarkiatasolle. Hierarkian alemmille tasoille annetaan toimintaohjeita mallin ylläpitämistä ja toiminnan suoritusta varten. Nykyiset robottijärjestelmät käsittelevät tasojen 0, 1 ja 2 mukaista informaatiota.

6.0 OHJELMOINTI

Ohjelmoinnin tarkoituksena on saattaa tuotteen kokoonpano-ohje robotin ohjausjärjestelmän ymmärtämään muotoon. Robottijärjestelmä tulee ohjelmoida aina uudelleen, kun tuotteisiin tulee muutoksia. Tyypillisesti kulutushyödykkeiden elinaika on hyvin lyhyt; uusia malleja ilmestyy markkinoille usein. Tästä seuraa jatkuva tuotantokoneiston muutostarve. Robottituotannossa tämä merkitsee jatkuvaa ohjelmankehitystä, mikä korostaa ohjelmointimenehtelmän joustavuuden tärkeyttä.

Ohjelmointi jakautuu eri vaiheisiin siten, että ensin määritellään työkierron kulku eli kokoonpanosekvenssi, joka toteutetaan käytettävissä olevalla robottilaitteistolla. Robotin ohjausohjelman tehtävänä on pitää kokoonpanon kulku haluttuna.

6.1 TYÖKIERRON SUUNNITTELU

Työkierron suunnittelu muodostaa varsinaista robotin ohjelmointia edeltävän suunnitteluvaiheen, jossa määritellään mm. käsiteltävien kappaleiden sijoittelu työalueelle, kappaleiden kokoonpanojärjestys, työmenetelmät, robotin liikeradat sekä turvalliset tarttumiskohdat. Mikäli tämä työvaihe on riittävän hyvin suunniteltu, muodostaa robottiohjelman koodaus enää suhteellisen pienen osan kokonaistyömäärästä.

6.1.1 Törmäysten esto

Robotin toimiessa kokoonpanotehtävissä voi robotti törmätä vieraisiin työkappaleisiin käsivarren liikkeiden, sekä käsiteltävään työkappaleeseen tarttumisen aikana. Käytettäessä kahden robottikäsivarren järjestelmää voivat käsivarret törmätä myös toisiinsa.

Törmäysten estossa on liikkeiden täydellinen kontrolli välttämätöntä. Käytettäessä suoraviivaisia liikkeitä toteutetaan robotin käsivarren liikkeet ennakoitavalla tavalla, mutta jos käytetään koordinoimatonta ohjausta, ei robotin liikerata ole määritelty, vaan se määräytyy robotin kulloisenkin alkuaseman mukaan.

Robotin liikkuessa ympäristössä, jossa on muita kiinteitä kappaleita on törmäysvaara aina olemassa. Mahdollisen törmäyksen seuraukset ovat useimmiten kohtalokkaita, joten törmäysten välttämiseen on kiinnitettävä erityistä huomiota.

Yleisin tilanne kokoonpanotyössä on se, että työkappale muodostaa törmäysvaaran. Törmäykset voidaan välttää joko liikkeiden huolellisella suunnittelulla ja simuloinneilla tai käyttämällä aina törmäyksistä vapaata kulkureittiä.

Koottaessa kappaletta ylhäältä alaspäin voidaan usein olettaa, että tietyn Z-tason yläpuolella ei enää ole esteitä. Siirtymiset turvalliselle työskentelyalueelle hidastavat toimintaa merkittävästi, mutta on käyttökelpoinen ratkaisu työkierron suunnittelun alkuvaiheessa. Lisäksi periaate on joustava ylläpidon kannalta.

Joskus on kahden robottikäsivarren yhtäaikainen käyttö välttämätöntä. Esimerkiksi piirilevyn testauksessa, jolloin käsivarsiin on asennettu koestuselektrodit. Kokoonpanossakin tarvittaisiin usein kahden käsivarren yhteistyötä, jolloin käsivarsien törmäysvaara on olemassa. Mikäli käsivarsien työskentelyalueet peittävät toisensa ja käsivarsilla on useita vapausasteita, tulee automaattinen, mallipohjainen törmäystarkastus hyvin raskaaksi, hitaaksi ja tietojenkäsittelyresursseja kuluttavaksi.

Törmäystarkastelua yksinkertaistaa huomattavasti se, että käytetään sellaisia robottirakenteita, joissa tarkastelu suoritetaan vain yhden vapausasteen suunnassa. Tällainen rakenne saavutetaan sijoittamalla porttalirakenteisen robotin kaksi käsivartta samalle pituusakselille /9/.

6.1.2 Kiertoajan minimointi

Työkierron optimoinnilla pyritään työkierron mahdollisimman nopeaan suoritukseen. Suoritusajatarkastelut voivat perustua ihmistyövoimalla tehtyihin tutkimuksiin, joissa oletetaan, että käytettävissä on tietty joukko standardityöliikkeitä, joiden optimaalinen järjestys takaa nopeimman suorituksen (MTM 1. methods-time-measurement). Robottia ei kuitenkaan voi suoraan verrata ihmiseen; suurimmat eroavaisuudet ovat kouran rakenne ja käsivarsien lukumäärä. RTM (robot-time-motion) analyysi perustuu edellisen kaltaiselle periaatteelle, jossa optimoinnin pohjana käytetään robotin ominaisuuksia kuten maksimi vääntömomentti, erotuskyky, vapausasteet ym. Optimointiohjelmaa ei kuitenkaan sisällytetä robotin ohjausyksikköön, vaan se toimii suuremmassa tietokoneessa esim. CAD/CAM-järjestelmän osana.

Optimointiohjelmat ovat raskaita, ja siksi niiden käyttö teollisuudessa on vähäistä. Enemmän käyttöä on robottisimulaattoreiden aikaa mittaavilla ominaisuuksilla, joilla eri kokoonpanomenetelmiä ja liikeratoja voidaan vertailla. Esimerkiksi IBM:n CAD/CAM-järjestelmään kuuluvalla CATIA-ohjelmistolla voidaan simuloida käyttäjän määrittelemän robotin toimintaa graafisen työaseman avulla. Kun simulointiin

yhdistetään mallit työaseman oheislaitteista ja valmistettavaa tuotteesta, saadaan varsin todenmukainen kuva kokoonpanon suorituksesta. Kaikki kappaleet esitetään volyymimalleina, minkä etuna on piiloviivojen automaattinen poisto. Tämä on välttämätöntä käsiteltäessä monimutkaisia kappaleita, joita asetellaan toistensa yhteyteen. Tietokoneiden tehon kasvaessa voidaan liikeratojen simulointi saada tarkasti vastaamaan todellisuutta.

Käyttäjä voi luoda valmistettavien kappaleiden lisäksi kirjaston käytettävissä olevista robottityypeistä, joten ohjelmistoa voidaan käyttää myös eri robottien vertailuun. CATIA-ohjelmiston robotiikkamoduli ei luo varsinaista robotin ohjausohjelmaa, mutta siitä saatavaa tietoa voidaan käyttää apuna määriteltäessä kokoonpanossa käytettäviä liikeratoja, asennussuuntia, tarttumiskohtia ym.

6.2 ON-LINE-OHJELMOINTIYMPÄRISTÖ

On-line-ohjelmoinnissa käytetään todellista tuotantokäyttöön soveltuvaa robottia joko laboratoriossa tai tuotantoympäristössä. Robotin liikkeet voidaan opettaa näyttämällä tai ohjelmoimalla robottia interaktiivisesti käsky kerrallaan. Valmis ohjelmakoodi tulee samalla testatuksi ympäristössä, joka muistuttaa riittävän tarkasti lopullista tuotantoympäristöä, joten robotin ohjausohjelma voidaan ottaa välittömästi ohjelmoinnin jälkeen käyttöön.

Koska robotti on ohjelmoinnin ajan pois tuotantokäytöstä, on edullista minimoida ohjelmointiaika. Käyttämällä yksinkertaista ja nopeaa opetusmenetelmää voidaan ohjelmointiaikaa lyhentää, mutta kokoonpanosovelluksissa tämä on harvoin mahdollista, sillä mukautuminen ympäristön muutoksiin ja epätarkkuuksiin vaatii kehittyneitä ohjelmistoja, joita näyttäen opettamalla ei voida saada aikaan. Käytännön ratkaisuna on kehittää ohjelmistoa erillisen robottisimulaattorin avulla off-line-ympäristössä.

6.3 OFF-LINE-OHJELMOINTIYMPÄRISTÖ

Off-line-ympäristössä ohjelmointi tapahtuu tietokoneavusteisesti täysin erillään tuotantoympäristöstä. Off-line-ympäristö on usein graafinen työasema, jossa operoidaan graafisella kolmidimensioisella mallilla. Graafinen malli simuloi robottia, sen työympäristöä ja manipuloitavia kappaleita. Malli voidaan luoda CAD-järjestelmällä, tai se voi sisältyä itse ohjelmaan, mikäli tullaan toimeen hyvin yksinkertaisella mallilla, joka voisi muodostua esim. kappa-

leiden tarttumapisteistä. Robotin malli on vakio, ja se voi olla simulaattorijärjestelmässä valmiina.

Ohjelmankehitys etenee off-line-ympäristössä samalla tavoin, kuin todellisessa ympäristössä. Työalue ja käsiteltävät kappaleet voidaan suunnitella CAD-järjestelmällä. Robotin liikeradat voidaan niinkään suunnitella ja testata samaan järjestelmään kuuluvan simulaattorin avulla.

Simulaattorin robottimallin tulee vastata mahdollisimman tarkasti todellisen robotin toimintoja. Tästä seuraa se, että yksinkertaisen robotin simulointi on helppoa ja se voidaan suorittaa pienelläkin tietokonelaitteistolla.

Esimerkkinä IBM 7535- robotin simulointi henkilökohtaisella tietokoneella. Simuloinnin onnistuminen perustuu robotin vapausasteiden vähäisyyteen ja vakioihin toimintaperiaatteisiin. Robotilla on neljä vapausastetta ja sen suorittamat kappaleenkäsittelytoimet tapahtuvat aina suoraan pystysuunnassa, joten kaksiulotteinen geometrinen malli riittää hyvin kuvaamaan robotin kaikkia toimintoja.

Ohjelmankehitys voidaan jakaa eri vaiheisiin, joiden työmäärästä n.90% voidaan tehdä off-line-ympäristössä (Kuva 14 sivulla 38).

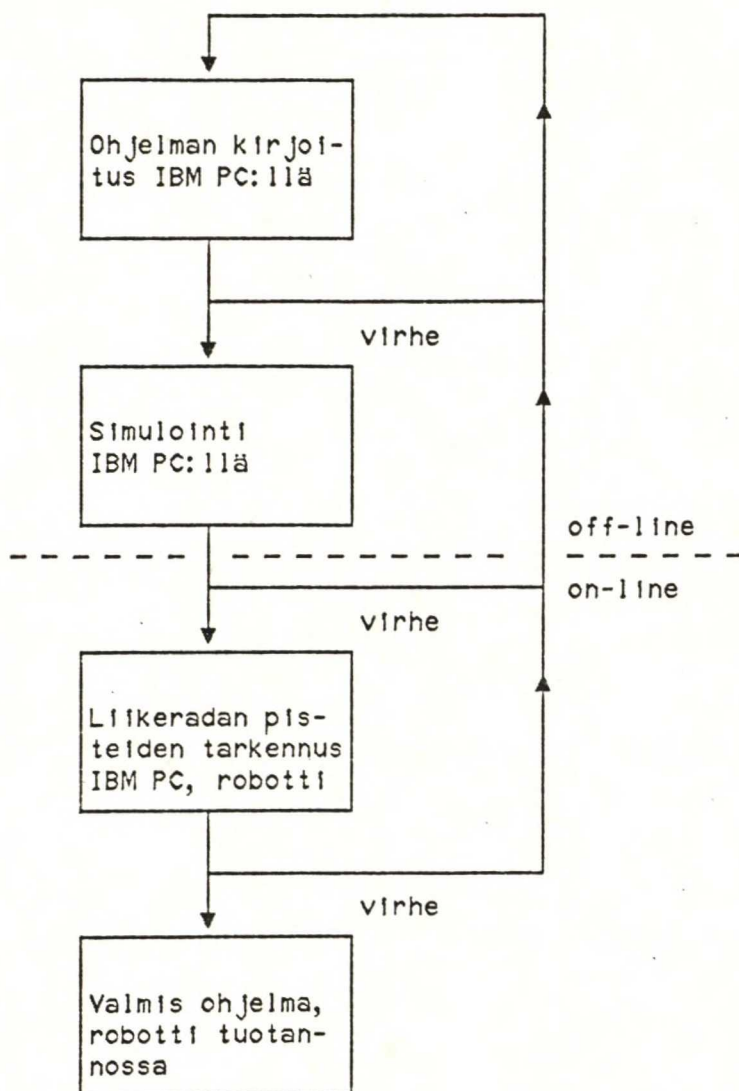
Kun työkappaleen malli on saatu valmiiksi ja kokoonpanoprosessin kulku on selvitetty, on tarvittavat liikeradat määriteltävä simulointijärjestelmän avulla. Robotin liikkeisiin kuluva aikaa voidaan laskea ja näin optimoida liikeratoja. Ohjelman koodaus ja lataus suoritetaan simulointijärjestelmällä, joten erillistä robottikohtaista ohjelmointijärjestelmää ei tällöin välttämättä tarvita. Näin robotin ohjausyksiköstä voidaan tehdä varsin yksinkertainen.

Vaikka ohjelma voidaankin kehittää off-line-ympäristössä, esiintyy roboteissa yksilöllisiä eroavaisuuksia, joten ohjelmat eivät sellaisenaan ole valmiita tuotantokäyttöön. Ratkaisuna on näyttää tarkat pisteet erikseen lyhyessä on-line-ohjelmointijaksossa, tai käyttää sensoreita eliminoimaan pienten epätarkkuuksien aiheuttamat virhetilanteet.

Mikäli robotin toiminta ohjautuu ympäristön mukaan esimerkiksi sensoreiden avulla, on robottijärjestelmän täydellinen simulointi lähes mahdotonta, sillä ympäristön satunnaisia epätarkkuuksia on erittäin vaikeata kuvata täsmällisessä tietokoneympäristössä.

6.4 ERI OHJELMOINTIMENETELMÄT

Ohjelmointimenetelmän soveltuvuus kohteeseen riippuu tehtävän ja ohjelmankehitysympäristön luonteesta, sekä



Kuva 14. Ohjelmankehitys IBM 7535-simulaattorilla /7/

käyttäjän taidoista. Käyttäjäliittymän on oltava sellainen, että kokoonpanotehtävän teoriassa hallitseva henkilö kykenee ohjelmoimaan sen robotille. Kehittyneissä ohjelmointijärjestelmissä voi olla useita eritasoisia liittymäjä sen mukaan, kuinka paljon kokemusta käyttäjällä on.

Mahdollisia robotin käyttäjiä/10/:

- Operaattori, joka kommunikoi robotin kanssa painonap-pien, merkkivalojen sekä päätteeltä esitettyjen kysymys-ten välityksellä.

- Ylläpito- tai huoltohenkilö, jonka tehtävänä on ajaa huolto-ohjelmia tai ohjelmoida lyhyitä testiohjelmia.
- Sovellusohjelmiston käyttäjä, joka kykenee muokkaamaan standardiohjelmia ja generoimaan omia sovelluksia.
- Sovelluskirjastojen ohjelmoija, joka kykenee suunnittelemaan ja ohjelmoimaan yleiskäyttöisiä ohjelmakirjastoja.

6.4.1 Opetus näyttämällä

Opetusmenetelmä perustuu siihen, että robotin liikkeet opetetaan liikuttamalla robotin käsivartta työkierron mukaisesti haluttujen pisteiden kautta. Yksittäisessä pisteessä voidaan ohjata esim. tarttujan liikettä tai käsitellä liitäntöjä ulospäin. Valmis ohjelma muodostaa jäykän sekvenssin, jota robotin käsivarsi toistaa./12/

Menetelmän etuna on havainnollisuus ja käytön helppous. Ohjelmointiin ei tarvita ohjelmoinnin asiantuntijaa. Ohjelman suorittamiseen ei tarvita yleiskäyttöistä tietokonetta.

Haittapuolena on se, että ohjelman suoritus ei voi haarautua sensoreista johdetun informaation mukaan. Menetelmä on hyvin kömpelö, mikäli samaa liikesarjaa on toistettava eri paikoissa. Opetusmenetelmä soveltuu maalaukseen ja piste-hitsaukseen, mutta ei kokoonpanoon, jossa ympäristöön mukautuminen on keskeistä, sillä tarvittavien toimintatapojen ohjelmointi näyttämällä on erittäin vaikeaa. Lisäksi ohjelman editointimahdollisuudet ovat rajoitetut, mikä vaikeuttaa dokumentointia ja ylläpitoa.

Opetusmenetelmään on kehitetty lisäominaisuuksia, joiden avulla osa epäkohdista vältetään. Tällaisia ovat ns. työkalukoordinaatiston käyttömahdollisuus, liikkeen pysäyttäminen sensoreista johdetun informaation mukaan, ehdollinen ohjelman haarautuminen, ja näköjärjestelmän liittämissämahdollisuus.

Ohjelmointi näyttämällä soveltuu kohteisiin, joissa liikeraata tai tietyn paikan saavuttaminen on tärkeämpää, kuin ko. paikassa tehtävä työ. Vaikka kehittyneimmissä kielissä onkin parannuksia, puuttuu niistä kuitenkin korkean tason ohjelmointikielten joustavuus ja avoimuus.

Mikäli kokoonpanossa esiintyy tarkkoja sovituksia, suuria paikan epävarmuuksia, viallisia tai väriä komponentteja, ei tarvittavia virheenkorjausrutiineja voida ohjelmoida näyttämällä. Tietyillä mekaanisilla mukautuvilla rakenteilla voidaan osa esiintyvistä ongelmista ratkaista.

6.4.2 Robottitasoinen ohjelmointi

Näyttämällä ohjelmoinnissa ei kyetä hallitsemaan kaikkia tarpeellisia oheistoimintoja. Käytännössä on havaittu, että tietyissä sovelluksissa kokonaisohjelmakoodista vain noin 10% liittyy robotin liikutteluun. Loput 90% ohjelmistosta on käytetty mm. käyttäjäliittymän, kalibrointirutiinien, virheen käsittelyrutiinien sekä tarvittavien laskentatehtävien toteuttamiseen /10/. Tämän takia tarvitaan korkean tason kieliä, kuten Pascal ja Basic, muistuttavia robotin ohjelmointikieliä, joiden avulla kyetään hallitsemaan tarvittavat tietojenkäsittelyrutiinit.

Robottitasoisissa ohjelmoinnissa operoidaan robotin rakenteeseen liittyvillä käsitteillä kuten tarttumakouran asemalla ja liikkeellä, sensoreiden tuntemilla voimilla yms. Ohjelmointikielissä on toteutettu normaalit kontrollirakenteet, kuten IF... THEN... ELSE, ja aliohjelmakutsut, joten strukturoitu ohjelmointi on mahdollista.

Ongelmana robottitasoisten ohjelmointikielten käytössä on kielten erilaisuus. Jokaisella robottityypillä on oma ohjelmointikielensä. Standardin, robotin rakenteesta riippumattoman ohjelmointikielen kehittäminen edistäisi ohjelmointimenetelmän yleistymistä. Käyttäjän kannalta ongelmana voi olla se, että hyvän ohjelmoijan tulisi olla asiantuntija sekä ohjelmointitekniikassa että manipulaatio-ritekniikassa /5/.

6.4.2.1 AML-ohjelmointikieli

IBM 7565-robotin ohjelmointikieli on AML (A Manufacturing Language), jonka suunnittelussa on otettu erityisesti huomioon eritasoisten käyttäjien erilaiset tarpeet ja ohjelmointikokemus. AML on ohjelmointikieli, jonka ominaisuudet kattavat korkean tason ohjelmointikielen funktiot, ja joka sen lisäksi soveltuu joustavasti robottiohjelman kehitykseen ja robotin ohjaukseen.

AML-kieli tuntee kokonaisluvut, reaalitylvut, merkkijonomuuttujat sekä erityiset vektorimuuttujat, joita AML:ssä kutsutaan aggregaateiksi. Näiden vektorimuuttujien alkioina voi olla kaikkia edelläkuvattuja tietotyyppjä tai toisia aggregaatteja. Muuttujia voidaan manipuloida tavanomaisten aritmeettisten operaatioiden kuten yhteen-, vähennys-, kerto- ja jakolaskun avulla. Loogiset operaatiot kuten AND ja OR ovat käytössä samoin kuin vertailuoperaatiot LT, LE, EQ, ym. AML-kielessä käytetyt kontrollirakenteet ovat IF..THEN..ELSE, WHILE..DO, REPEAT..UNTIL, BEGIN..END sekä SUBR..END.

AML-kieli rakentuu aliohjelmista, jotka voivat kutsua toisiaan. Muodollisesti aliohjelmat eivät eroa toisistaan. Systeemialiohjelmat ja käyttäjän luomat aliohjelmat ovat myös liitännöiltään samanmutoisia.

Ohjelmointikielessä on komennot oheislaitteiden hallintaa varten sekä luonnollisesti joukko komentoja robotin ohjausta varten. Robotin ohjaukskomentoja ovat mm. MOVE, DMOVE, GUIDE, SPEED, MONITOR. GUIDE-komennon avulla voidaan AML-ohjelmaan sisällyttää näyttämällä opetettuja ohjelmaloheja.

AML-kielellä on toteutettu käyttäjäystävällisiä liityntöjä, joiden avulla myös vähäisen ohjelmointikokemuksen omaava henkilö voi näyttämällä opettamalla tuottaa käyttökelpoisia ohjelmia ja kommunikoida robotin kanssa /10/. Tämä on tärkeää käytettäessä robotteja tuotantoympäristössä, jossa luonnostaan ei ole ohjelmointikokemusta.

Ohjelmointikieli on tulkkityyppinen, sillä näin saadaan toteutettua interaktiivinen ohjelmointiympäristö ja mahdollisten virheiden korjaus helpottuu ja nopeutuu. Tulkin suhteellinen hitaus verrattuna käännettyjen ohjelmien suoritukseen ei tavallisesti aiheuta ongelmia, sillä robotin liikkeiden ohjaus ei vaadi suurta laskentanopeutta. Mahdolliset nopeusongelmat voidaan kiertää käyttämällä etukäteen käännettyjä moduleita nopeutta vaativiin tehtäviin. Ohjelmatulkki on myös vastaavantasoiseen kääntäjään verrattuna yksinkertaisempi toteuttaa.

AML-ohjelmointiympäristössä on mahdollista korjata mahdolliset virheet välittömästi ja jatkaa toimintaa siitä, mihin jäätiin seuraavien kommentojen avulla:

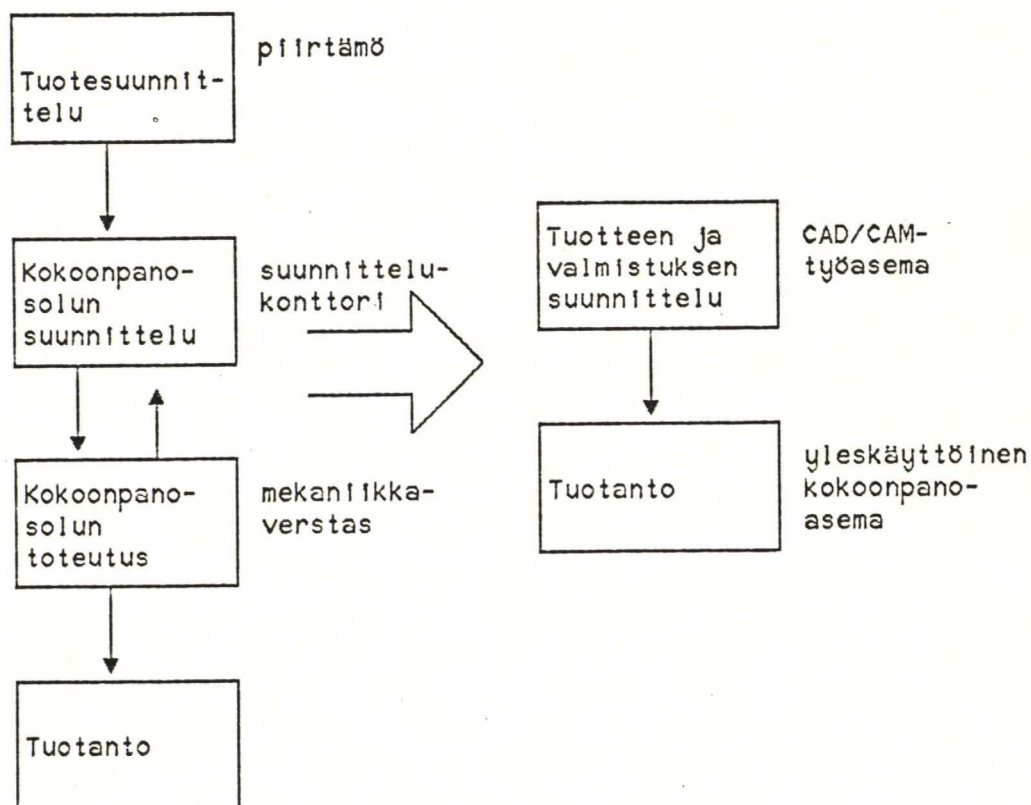
TRACE	Tulostaa suoritettut käskyt päätteelle, kirjoittimelle tai tiedostoon.
LINESTOP	Mahdollistaa ohjelman etenemisen käsky kerrallaan.
BREAK	Tulostaa suoritettun koodin ja mahdollistaa AML-ohjelman muokkaamisen.
ATTENTION	Keskeyttää välittömästi ohjelman suorituksen.

Ohjelmat kehitetään useimmiten on-line-ympäristössä. Ohjelmankehitys sitoo näin ainakin yhden robottilaitteiston. Sensoripohjaisen AML-ohjelman simulointia varten on kehitetty kokeilujärjestelmä /2/. Ohjelma sisältää sensoreihin perustuvia ehdollisia haarautumisia. Sensoreita simuloidaan mallin perusteella tai interaktiivisesti.

6.4.3 Tehtävätasoinen ohjelmointi

Robotilla suoritettavien tehtävien tullessa monimutkaisemmiksi on syntynyt tarve nostaa ohjelmoinnin tuottavuutta. Tavoitteena on, että tuotesuunnittelun yhteydessä ja sen aikana syntyisi myös tuotteen valmistuksessa tarvittavat tiedot, jolloin valmistuksen suunnittelu yksinkertaistuisi ja viive suunnittelun ja valmistuksen välillä lyhenisi (Kuva 15).

Ohjelmointimenetelmien kehittämiseksi on tutkittu mahdollisuuksia hyödyntää CAD-tietokantaan talletettuja tuotetietoja yhdistettyinä korkean tason kokoonpano-ohjeisiin. Näin on kokoonpanossa tarvittavat asiat määritelty ja edellytykset automaattiseen robottitasoisen ohjelman generoimiseen ovat olemassa.



Kuva 15. Suunnittelun integrointi

Tehtävätasoinen ohjelmointikieli käsittelee asennustehtävää eikä puutu siihen, millä tavalla tehtävä suoritetaan.

Esim. INSERT tappi IN reikä.

Ohjelmointimenetelmän käyttö perustuu siihen olettamukseen, että kokoonpanotehtävissä esiintyvät ongelmat voidaan ratkaista automaattisesti, kunhan robotilla on tarvittavat ominaisuudet. Näin voidaan ohjelmoida huomattavan pitkiä ja monimutkaisia kokoonpanoprosesseja, ja dokumentoinnista tulee automaattisesti selkeää.

Tehtävätasaisen järjestelmän tulee tuottaa robottitasoista koodia, johon on voitava vielä lisätä virheestötoipumisrutiineja tarpeen mukaan. Ennustettavissa olevat virhetilanteet tulee käsitellä sensoritietoja käyttävillä rutiineilla.

6.4.3.1 Käsitteet, komponentit

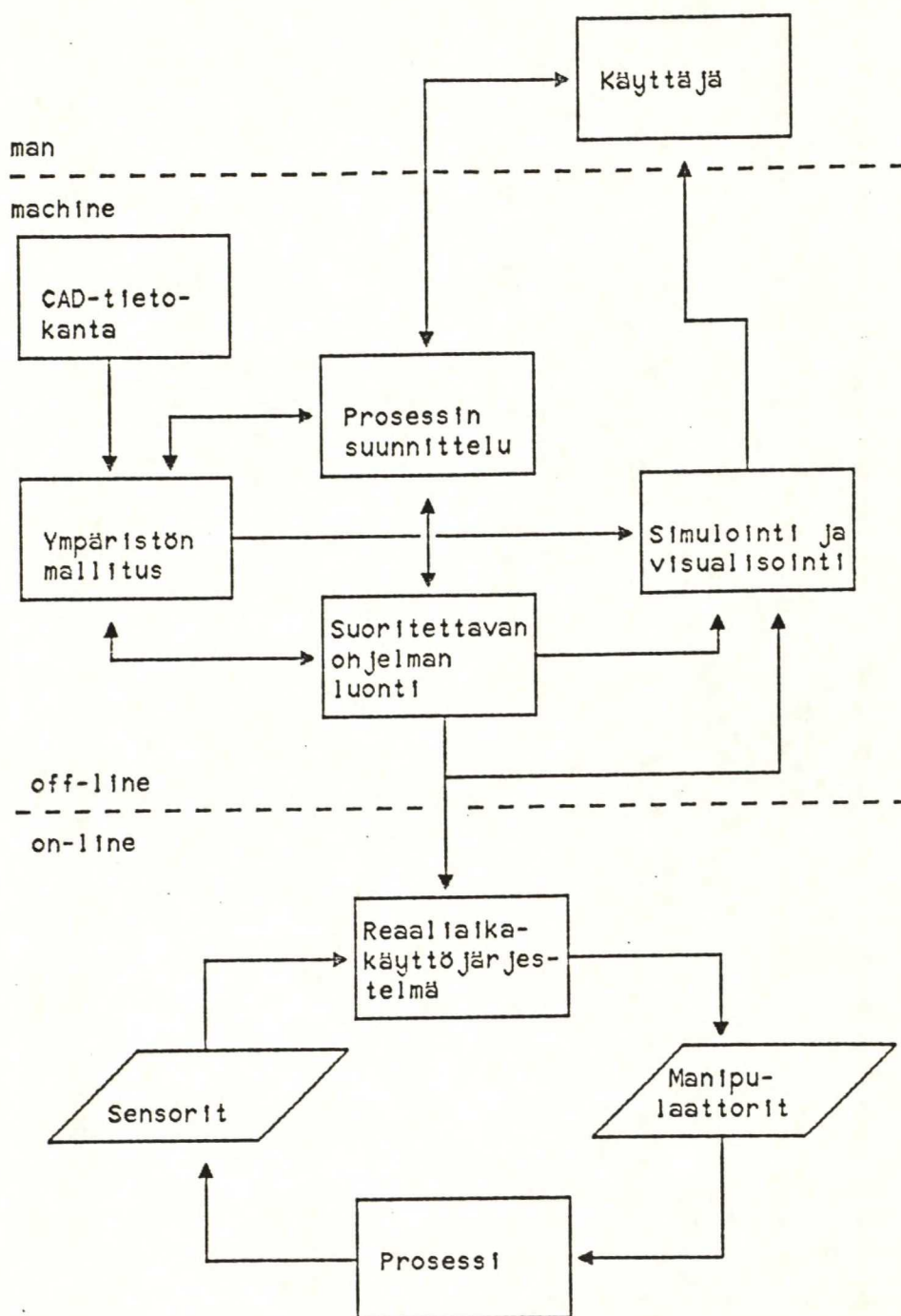
Robottitasoisen ohjelman automaattista generointia varten tarvittava järjestelmä voisi sisältää oheisen kaavion (Kuva 16 sivulla 44) mukaiset modulit. Tehtävätasosta ohjelmointia tukevat modulit ovat jo nykyisin olemassa, varsinaisen käännöstyön suorittavat modulit 'prosessin suunnittelu' ja 'suoritettavan ohjelman luonti' puuttuvat. Integrointityö ei myöskään ole vielä suoritettu, sillä eri modulien rajapintoja ei ole standardoitu.

Ohjelmointityö etenee siten, että ensin määritellään ympäristön malli. Mallia käyttäen laaditaan toiminnallinen kokoonpano-ohje, jossa kokoonpanon edistymisen kuvataan tilojen avulla. Esimerkiksi ruuvi A on kierretty reikään B. Ohjelmointijärjestelmän ehdottama robottitasoinen ratkaisu tarkastetaan simuloinnein ja tarvittaessa annetaan lisämäärittelyjä. Käyttäjän hyväksymää ohjelmaa voidaan vielä muokata robotin toimintaympäristössä.

6.4.3.2 Ympäristön mallitus

Kokoonpanotehtävä kuvataan sarjana malleja, jotka kuvaavat kokoonpanon eri vaiheita tilasta tilaan. Jotta kokoonpanon suunnittelu voisi onnistua pelkkään malliin tukeutuen, olisi mallin sisällettävä tiedot kappaleiden geometriasta, fyysikaalisista ominaisuuksista sekä kappaleiden välisistä kineemaattisista suhteista /12/. Käsiteltävien kappaleiden mallien lisäksi myös robotin malli on sisällytettävä järjestelmään.

Geometrinen malli saadaan tietokoneavusteisen suunnitteluprosessin tuloksena. Sen tulee sisältää tiedot kaikkien kappaleiden muodoista tilavuusmallina. Viiva- ja pintamallit eivät sisällä riittävää informaatiota tehtävänsuunnittelijan



Kuva 16. Tehtävätasoinen ohjelmointiympäristö /19/

tarpeisiin /12/. Toleranssitietoja tarvitaan mukautumis- ja virheestätoipumisstrategioiden suunnitteluun.

Fysikaalinen malli sisältää kappaleiden ne fysikaaliset ominaisuudet, jotka ovat välttämättömiä kokoonpanon kannalta.

Massa , hitausmomentti ja kitkakerroin on näistä tärkeimmät. Esimerkiksi pienin mahdollinen tarttumisvoima voidaan määrittää kappaleen massan ja kitkakertoimen avulla.

Kinemaattinen malli kaikista liittymäkohdista sisältää tiedot kappaleiden välisten liittymäkohtien vapausasteista, joiden perusteella sallitut liikesuunnat määräytyvät. Kinemaattisen mallin sisältämä informaatio muuttuu jatkuvasti kokoonpanon edistyessä.

Robotin malli voisi koostua seuraavasti:

- Robotin kinemaattinen malli. Robotin nivelten rajat
- Robotin dynaaminen malli. Suurimmat käytettävissä olevat kiihtyvyydet
- Sensoreiden mallitus, ja niiden käyttö

Robotin monimutkainen malli on luotava vain yhden kerran, jonka jälkeen mallia voi käyttää moniin eri sovelluksiin. Erilaisista robottirakenteista voidaan muodostaa vapaasti käytettävissä oleva kirjasto. Ympäristön malli on sen sijaan luotava jokaista sovellusta varten erikseen. Yksittäisistä kappaleista saadaan mallitustiedot tuotteen suunnitteluvaiheen tuloksena. Muussa tapauksessa mallin generoimiseen kuluu niin paljon työtä, että robottitasoisen ohjelman luonti manuaalisesti on nopeampaa.

Mallin yksityiskohtien lukumäärä määräytyy tehtävän vaikeustason mukaan eikä kaikista yksityiskohdista tarvita täydellistä mallia. Mallia voi tarkentaa, mikäli tehtävän suoritus sitä edellyttää.

6.4.3.3 Prosessin suunnittelu

Kokoonpanotyön edistyminen esitetään tilajonona. Tilojen lukumäärä riippuu tehtävänmäärittelijän kehittyneisyydestä. Paras tilanne olisi sellainen, jossa vain kokoonpanotyön alku- ja lopputila pitäisi määritellä.

Kokoonpanotyö on työlästä kuvata sarjana täydellisiä malleja, sillä geometrinen tai kinemaattinen mallikaan ei aina ole riittävä, jos esim. on kiristettävä pultti tiettyyn kireyteen. Huomattavasti yksinkertaisempaa on kuvata ainoastaan operaatiot, joiden perusteella mallia päivitetään. Tämän takia operaatio-orientoituneet tehtävän ratkaisut ovat antaneet tutkimuksissa lupaavimpia tuloksia /12/.

Tehtävän määrittelyyn voidaan käyttää seuraavanlaisia komentoja: PICKUP(esine), MOVETO(paikka).

Tässä vaiheessa eivät robotin ominaisuudet vielä vaikuta suunnitteluun. Tuloksena ohje, joka määrää kokoonpanojärjestyksen ja menetelmät, joilla kappaleet yhdistetään toisiinsa.

6.4.3.4 Suoritetavan ohjelman luonti

Tämä moduli kääntää tehtäväorientoituneen ohjelman robottitasoiseksi käyttäen apuna robotin mallia. Robottitasoinen ohjelma käsittelee sensori-informaatiota, suorittaa sensoreilla valvottuja liikkeitä, sekä hallitsee mahdolliset virhetilanteet. Tehtävänsuunnittelija tarvitsee toimiakseen tiedot ympäristöstä, tehtävän alkutilan, sekä halutun lopputilan.

Koska järjestelmissä ei ole ns. tekoälyä, on kaikki toiminnot määriteltävä yksikäsitteisesti. Määrittely helpottuu, mikäli käytetään interaktiivista ohjelmointijärjestelmää, joka kyselee ohjelmoijalta määrittämättömät kohdat /5/.

Ohjelman generointi voidaan jakaa kolmeen osaan:

- Tarttumisen suunnittelu
- Liikeratojen suunnittelu
- Sensori-informaation käsittely

Tästä vaiheesta saatavaa robottitasoista ohjelmaa voidaan suoraan käyttää tietyn robotin ohjaukseen.

Tarttumisen suunnittelu on yksi avainoperaatiosta. On ratkaistava tarttumissuunta, josta ei synny törmäystä kappaleen ja kouran välillä. Stabiili tarttumiskohta on selvitettävä. Tarttumisen oltava riittävän tukeva. Tarttumisoperaatio ei saa lisätä paikkatiedon epävarmuutta.

Liikkeet voidaan jakaa vapaisiin, valvottuihin ja mukautuviin.

Vapaata liikettä käytetään kappaleen siirtämiseksi paikasta toiseen niin, ettei törmäyksiä synny. Liikeratojen laskennassa on otettava huomioon robotin dynamiikka, mikäli halutaan toimia suurilla kiihtyvyyksillä.

Valvottu liike tapahtuu lähellä kohdetta sensoreiden valvonassa siten, että liike pysähtyy, kun tietty ehto, tavallisesti kosketus, tapahtuu. Nyt ohjausjärjestelmä tietää, että robotin koura on kappaleen pinnalla, mutta takeita siitä, että ollaan oikeassa paikassa ei ole. Oikea paikka saadaan selville sensoreiden avulla tutkimalla kappaleen pinnan muotoja ja vertailemalla mittauksia kappaleen malliin. Kappale

sovitetaan omalle paikalleen käyttäen kappaleen geometriaan mukautuvaa ohjausta.

6.4.3.5 Simulointi ja visualisointi

Simuloimalla havainnollistetaan generoidun ohjelman ja mallien toimintaa graafisella päätteellä. Valmis tulos esitetään animaationa, jossa kokoonpano tapahtuu graafista mallia käyttäen.

Simuloinnissa käsiteltävät mekanismit:
Sensoritakaisinkytkentä, tarttuminen ja irrottaminen, painovoima. Tilavuusmalli, moottoreiden nopeudet, nivelten rajoitukset kuuluvat simulointijärjestelmän malliin.

Monitorointi: Liikkeiden rekisteröinti, valitut katsomiskulmat, yhteentörmäyksen ilmaisu.

6.4.3.6 AUTOPASS-järjestelmä

AUTOPASS on IBM:n kokeilujärjestelmä, joka kykenee toteuttamaan tehtävatasoista robottiohjelmaa geometrisen mallin perusteella /5/.

AUTOPASS-kieli käsittelee seuraavanlaisia lausekkeita:

PLACE kannatin IN teline SUCH THAT
kannatin.pohja CONTACTS teline.huippu

PLACE liitin ON kannatin SUCH THAT
liitin.pohja CONTACTS kannatin.huippu
AND liitin.reikä IS ALIGNED WITH
kannatin.reikä

AUTOPASS:n ratkaiseva puute on se, että se ei ota huomioon todellisuuden ja geometrisen mallin välisiä eroavaisuuksia. AUTOPASS ei siis kykene hyödyntämään sensoritietoa siten, että haluttu tilanne saavutetaan odottamattomista virheistä huolimatta.

6.4.4 Robottien ohjelmointimenetelmien kehityssuuntia

Ohjelmointimenetelmien nykytila on sellainen, että jokaisella robottityypillä on oma ohjelmointikielensä, jota muut

robotit eivät ymmärrä. Tämä seikka estää robottiohjelmistojen kehittämistä ja näin se myös jarruttaa robottien soveltamista. Lähiajan kehityksenä tullee olemaan robottitasoisen ohjelmointikielen standardoiminen, jolloin eri valmistajien robotteja voisi ohjelmoida samalla ohjelmointikielellä.

Tämän vaiheen jälkeen voidaan odottaa järjestelmiä robottitasoisen ohjelman automaattista generointia varten. Tekoälyjärjestelmä tullaan tulevaisuudessa soveltamaan myös kokoonpanorobottien ohjaukseen, mutta vielä ei toimivia järjestelmiä ole kehitetty.

7.0 KÄYNNISTINKAAPIN KOKOAMINEN ROBOTILLA

Oy Wärtsilä Ab:n Turun telakan sähköpaja kokoaa laivojen sähkökäyttöjen kytkinkaappeja nykyisin täysin manuaalisesti työpajaympäristössä. Telakkateollisuuden kilpailun kiristyessä on telakan tuotantokustannuksia pienennettävä. Tämän vuoksi päätettiin tutkia robotin soveltuvuutta kytkinkaappien kokoonpanoon. Tätä varten oli selvitettävä kaikki muutokset, joita robotisointi toisi mukanaan sekä tuotteisiin että organisaatioon.

Paras tapa tutkia robotin soveltuvuutta tiettyyn kohteeseen on tehdä toimintakoe valmiilla peruslaitteistolla. Sähkökaappien valmistusta tutkittiin IBM RS/1-robotilla, joka on IBM 7565-robottia edeltänyt ns. nollasarjan malli, joka on ominaisuuksiltaan sama, kuin seuraajansa. Koe toteutettiin Ruotsin IBM:n Järfällassa sijaitsevan kirjotintehtaan robottilaitteistolla, joka oli varustettu pien-sarjatuotannon edellyttämällä kuljettimilla ja työkaluilla.

Kokeen tuloksina odotettiin saatavan tietoa siitä, mitä muutoksia olemassa oleviin tuotteisiin ja menetelmiin tulee tehdä, jotta automatisointi onnistuisi. Robotisoinnin myötä tultaisiin jatkossa saamaan kokemuksia robottiteknologian soveltamisesta aiemmin täysin ihmistyövoimavaltaiseen tuotantoon ja robotin vaikutuksista tuotantoprosessin muihinkin vaiheisiin.

7.1 LÄHTÖTILANNE

7.1.1 Nykyinen tuotanto

Käynnistinkaappeja valmistavan sähköpajan kokonaisvuosituotanto on n. 6000 erilaista kytkinkaappa, joita kokoaa n. 15-20 työntekijää työskennellen 8 tuntia päivässä. Tuotevalikoimassa on kuusi yleisesti esiintyvää variaatiota (Kuva 17 sivulla 50).

Tuote	vuosituotanto	sarjakoko	kalustusaika	johdotusaika
A	1300	100-200	1.5 h	5.5 h
B	500	100	2 h	12 h
C	150	20-30	1.5 h	0 h
D	300	20-30	1.5 h	0 h
E	3000		1 h	0 h
F	300		2 h	4 h
Muut	200			

Käytetty työaika yhteensä: 7225 h 14350 h

Kuva 17. Nykyiset tuotantomäärät

Tuotteen valmistus jakautuu kahteen päävaiheeseen: kytkinkaapin kalustukseen, ja kaapin johdotukseen. Näihin työvaiheisiin käytetyt ajat ilmenevät oheisesta taulukosta. Robottia tullaan ensi vaiheessa soveltamaan kytkinkaappien kalustukseen, sillä se on yksinkertaisempi tehtävä ja näin sen toteutuksen onnistumismahdollisuudet ovat myös suuremmat.

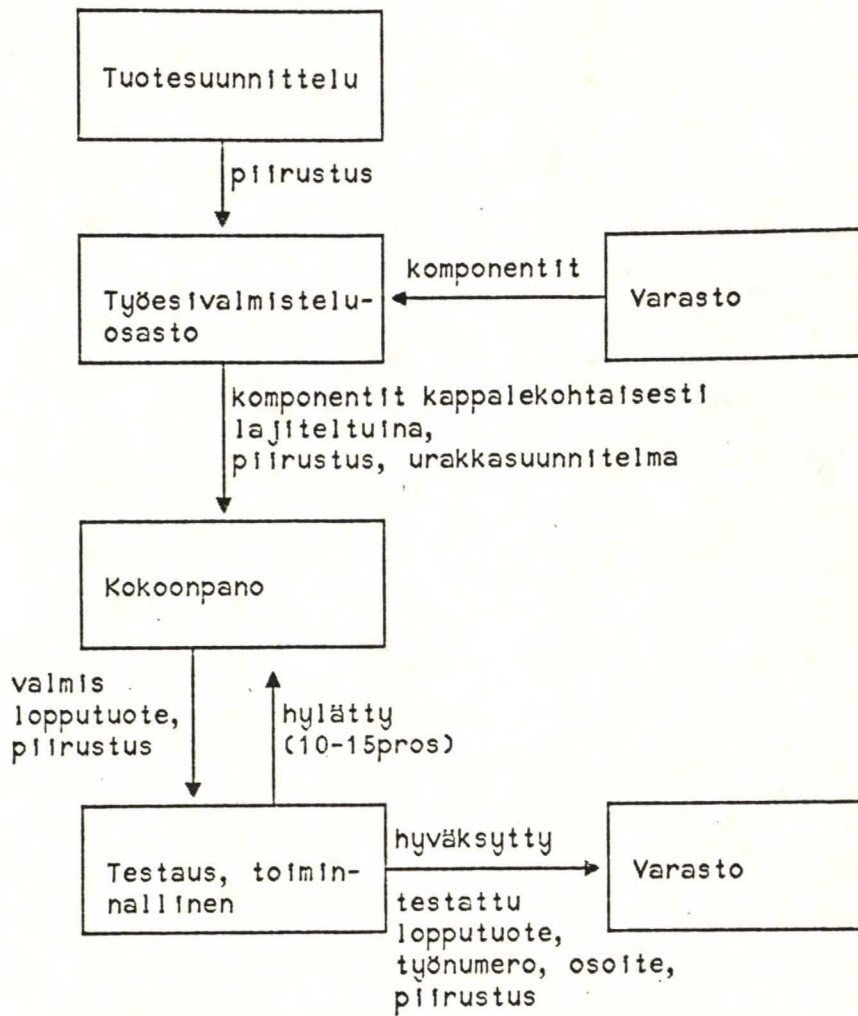
Kytkinkaapin komponentit kiinnitetään ruuveilla peltialustaan. Työvaiheet ovat:

- Kiinnitysreikien poraaminen
- Kiinnitysreikien kierteiden tekeminen
- Komponentin asettaminen paikalleen
- Komponenttien kiinnitys ruuvaamalla
- Salpakiinnitteisten komponenttien asennus

Robotille siirretään kolme viimeistä työvaihetta, sillä ne ovat ongelmallisimpia ja näin myös soveltuvuuskokeen kannalta oleellisia.

7.1.2 Materiaalin kulku

Materiaalin kulku kytkinkaappien manuaalisessa valmistuksessa on pääpiirteissään kaavion (Kuva 18 sivulla 51) mukainen.



Kuva 18. Materiaalin kulku kokoonpanoprosessissa

Kytlinkkaapit kootaan yksitellen valmiiksi tuotteeksi. Koska sarjakoko voi olla 100 kappaletta, on työn rationalisointitarve ilmeinen. Vaihtoehtoja on kaksi: joko rationalisoidaan valmistusprosessi siten, että kukin työntekijä suorittaa ainoastaan yhden työvaiheen, tai automatisoidaan valmistusta robottien avulla. Automatisointivaihtoehto tarjoaa lupavimmat mahdollisuudet, sillä todennäköisesti se nostaa myös muun tuotantokoneiston tehokkuutta ja tuotteiden laatua, jolloin epäsuorat kustannussäästöt tulevat olemaan huomattavat. Automaatiolla on ennen kaikkea mahdollista saavuttaa suurimmat säästöt valmistuksessa myös tulevaisuudessa.

7.2 ALUSTAVA KANNATTAVUUDEN ARVIOINTI

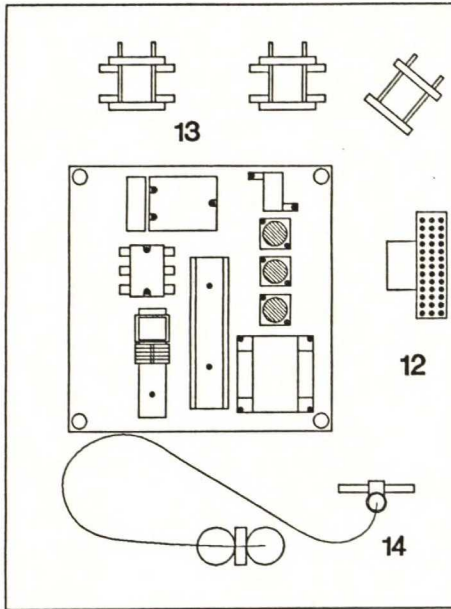
Koska kalustustyöstä noin puolet voidaan toteuttaa robotin avulla, näyttävät tuotantomäärät olevan riittävän suuria robotille ts. kehittynyt robotti saadaan tuottamaan täysipainoisesti ainakin yhdessä vuorossa, jolloin taloudelliset edellytykset robotin käytölle ovat olemassa.

Mikäli robotin käyttö tehostaa tuotantoa, on todennäköistä, että osaston tilauskanta kasvaa ja töitä robotille riittää myös toisessa ja kolmannessa vuorossa. Myös kannattavuus kasvaa vastaavasti. Kehittynyt ja monipuolinen robotti voi tulevaisuudessa kalustuksen lisäksi myös johdottaa kytkinkaapin, jolloin päästään tuottamaan täysautomaatioon ja ihmistyövoima vapautuu valmistamaan yksilöllisiä tilauksia, joiden valmistusta ei voida siirtää robotille. Tavoitteena on, että robotin käyttöönotossa ja tuotantoajossa kertyneet kokemukset siirretään yrityksen organisaation muihin osiin. Projektin koulutuksellinen merkitys on näin huomattava ja se tulee tehostamaan tulevaisuudessa toteutettavia robotisointiprojekteja.

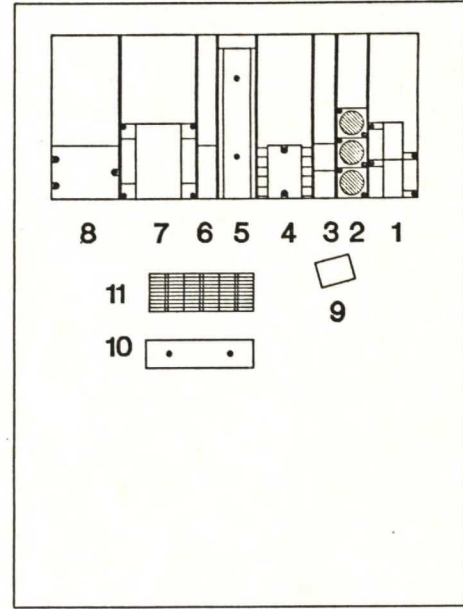
7.3 KOKOONPANOSOVELLUS IBM RS/1-ROBOTILLA

7.3.1 Laitteisto

Koejärjestelmän laitteiston rungon muodostaa IBM RS/1-robotti. Robotin työalueelle johtaa kaksi rinnakkaista rullarataa, joita pitkin tuodaan komponentit, työkalut sekä kokoonpanossa tarvittavat aputelineet. Robotin työalueella palleitit telakoidaan tarkasti paikoilleen, jolloin samalla kytkeytyy paineilma, käyttösähkö sekä signaalijohtimet robotin ohjaustietokoneen ja työkalujen välille. Työkalut: kolme tarttumapihtiä, ruuvitaltta ja levy riviliittimiä varten, ovat kaikki telineissään (Kuva 19 sivulla 53).



Työkalupalletti



Komponenttipalletti

- | | |
|---------------------------|-------------------------------|
| 1. Virtamuuntaja | 8. Kontaktori (suuri) |
| 2. Kolme sulakepesää | 9. Kontaktori (pieni) |
| 3. Kontaktorin lisäkytkin | 10. Kytkenäkiskoja |
| 4. Virtakytkin | 11. Riviliittimet |
| 5. Kaapelikouru | 12. Ruuviteline |
| 6. Lämpörele | 13. Telineet pihdeille |
| 7. Jännitemuuntaja | 14. Pneumaattinen ruuvitaltta |

Kuva 19. Komponenttien sijoittelu robotin työalueelle

Komponenttipalletilla on ennen kokoonpanoa alustalevy, johon kierteistetyt reiät on valmiiksi tehty. Lisäksi pallettiin on asennettu telineet kaksia pihtejä sekä ruuvitalttaa varten. Kullekin työkalulle on varattu oma telineensä, josta robotti voi poimia kulloinkin tarvittavan työkalun. Työkalujen vaihto tapahtuu näin vakio-ohjelmien avulla. Ruuvit poimitaan telineeseen poratuista rei'istä.

Työkalupalletilla sijaitsevat telineet komponentteja varten. Kuutiomaiset komponentit syötetään robotille kaltevia ratoja pitkin. Litteät kappaleet sijaitsevat pystytelineissä. Erilaisia komponentteja on yhteensä 11 kpl, joista osa kiinnitetään ruuvaamalla, ja osa salvan avulla aikaisemmin asennettuun komponenttiin. Ruuvit ovat kaikki 4 mm:n ruuveja.

Komponentti voi liikkua ruuvauksen ja muun työn aikana. Liikkumisen estäminen on yksi perusedellytys kokoonpanon

onnistumiselle. Tässä kokeessa käytettiin jousikuormitteisia tappeja, jotka olivat reikien kohdalla pitämässä komponenttia paikallaan ruuvauksen ajan. Ruuvien kiertyessä tappi painuu alas eikä häiritse ruuvausta.

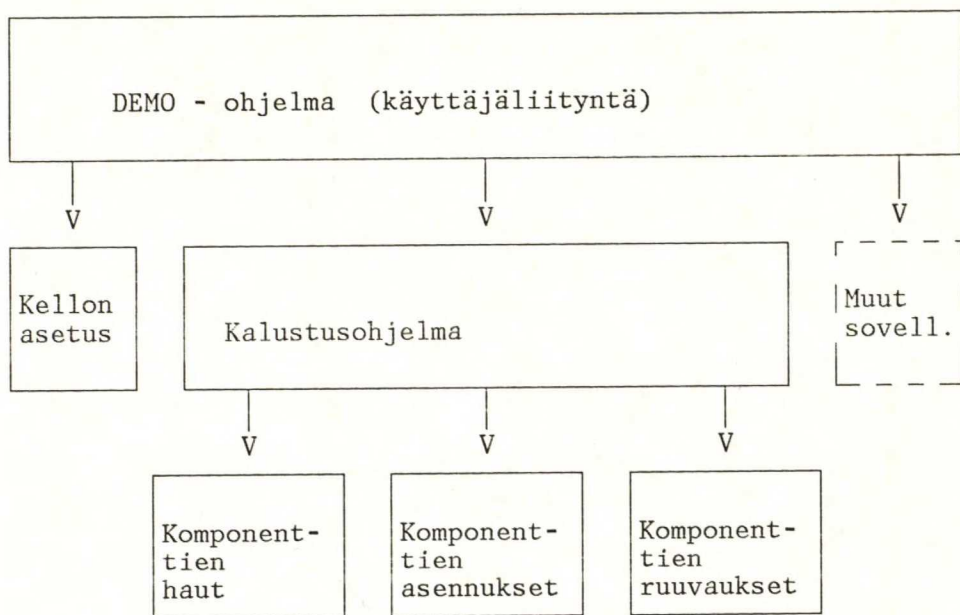
Ruuvaamiseen käytetään pneumaattista ruuvitalttaa, jossa pyörimissuunta, ruuvauspaine, vääntömomentti ja ruuvausnopeus ovat mekaanisesti aseteltavissa. Moottori pyörittää noin 10 cm pitkää ristipäällä varustettua ruuvitalttaa, jonka päälle asennetun putken avulla ruuvi imetään alipaineella taltan kärkeen. Ruuvi suoristuu kantansa ohjaamana, josta seuraa se, että kannan virheet ilmenevät ruuvien asettumisena vinoon. Ruuvitaltan käyttö mahdollistaa vielä sormien voimien valvomiseen.

Käsiteltävien komponenttien koko vaihtelee kolmesta senttimetristä kolmeentoista senttimetriin. Erikokoisten kappaleiden käsittelyyn käytetään kolmea erikokoista pihtiä. Pihtien kärjet on valmistettu kappalekohtaisesti siten, että niillä voidaan tarttua erimuotoisiin kappaleisiin. Työkalut ovat siten muotoillut, että robotti tarttuu kaikkiin näihin työkaluihin samalla otteella. Työkalut eivät häiritse tuntosensoreiden käyttöä, sillä ne kiinnittyvät robotin kouraan jousikuormitteisesti.

7.3.2 Sovellusohjelmisto

Ohjelmointi tapahtui AML-kielellä (A Manufacturing Language) /14/. Tulkkaava kieli mahdollisti tehokkaan ja joustavan ohjelmankehityksen. Ohjelmiston eri osat jaettiin omiin tiedostoihinsa, jotka ladattiin ja poistettiin työmuistista aina tarpeen mukaan, jolloin työmuistin koko ei rajoittanut ohjelmankehitystä. Esimerkiksi käyttäjäliittymän kuvaruututoiminnoissa käytettävät valikot ovat työmuistissa vain tarvittaessa.

Ohjelmisto rakentuu hierarkisesti siten, että käyttäjäliittymän toteuttava demo-ohjelma kutsuu haluttuja sovelluksia, joita tässä on järjestelmäkellon asetus sekä varsinainen robotin ohjausohjelma (Kuva 20 sivulla 55).



Kuva 20. Ohjelmiston rakenne

Käyttäjiliityntä-ohjelma DEMO (liite C, sivu 71) toimii siten, että kokoonpanossa normaalisti esiin tulevat toiminnot hallitaan valikkojen (liite D, sivut 85-89) avulla selväkielisiin kysymyksiin vastaamalla ja toimintaohjeita seuraamalla.

Valittavia sovelluksia ovat järjestelmäkellon asetus ja kalustusohjelma. Myös muita mahdollisia sovelluksia, esim. muiden tuotteiden kokoonpano-ohjelmia voidaan kutsua tästä valikosta. Kalustusohjelma muodostaa ohjelmiston tärkeimmän kokonaisuuden sisältäen kaikki vahvavirtakaapin kalustukseen liittyvät toiminnot.

7.3.3 Kalustusohjelma

Kaapin kalustusta ohjaava ohjelma SEQW (liite C, sivu 74) ohjaa kokoonpanoajärjestystä seuraavan järjestyksen mukaisesti: Pienten komponenttien asetus alustapellille, näiden ruuvaus kiinni, suurten komponenttien asetus, näiden ruuvaus, salpakiinniteisten komponenttien asennus. Kahdella ruuvausjaksolla haluttiin minimoida komponenttien liikkumisen alustalla muun työn aikana.

SEQW huolehtii siitä, että robotti ei törmää esteisiin toimintojen vaihtojen aikana. Komponenttien pystysuora asennussuunta mahdollistaa sen, että kouran törmäykset

työkappaleisiin voidaan vältetään käyttämällä vaakasuoriin siirtymisiin robotin työalueen vapaata yläosaa.

Kirjoittimelle tulostetaan työn aloitus ja lopetus sekä kokoonpanoon käytetty aika. Myös kaikista virheistä tulostuu ilmoitus kirjoittimelle. Mikäli kalustettavaan alustaan jää käyttäjän myöhemmin korjattavia vikoja, tulostuu näistä virheistä selvästi erottuva virheilmoitus. Kaikki ilmoitukset varustetaan kellonajalla jälkitarkastusta varten.

7.3.3.1 Komponenttien haut

Komponentit poimitaan telineistään TOKO-ohjelman ohjaamana (liite C, sivu 80). Ohjelma saa parametreina robotin asennon tarttumiskohdassa sekä kouran auki- ja kiinni-arvot. Näistä tiedoista ohjelma laskee liikeradat siten, että lähestyminen ja poistuminen tartuntapaikasta tapahtuu komponentin pystysuorien sivujen suuntaisesti. Ohjelma testaa lisäksi onko riittävä tarttumisvoima saavutettu. Mikäli voima ei ole riittävä, tulkitsee ohjelma tilanteen komponentin puuttumiseksi ja kehottaa käyttäjää lisäämään ko. komponentteja telineeseensä. Komponentin lisäyksen ja kuittauksen jälkeen ohjelma jatkaa normaalia toimintaansa.

7.3.3.2 Komponenttien asennukset

Suuri osa komponenteista asennetaan yksinkertaisesti laske-malla komponentti paikalleen. PLAKO-ohjelma suorittaa kappaleen laskemisen irrotuksen ja poistumisen kappaleen yläpuolelle (liite C, sivu 80). Kappaleen paikka annetaan parametreina ohjelmalle.

Sovitettavilla kappaleilla on kullakin oma sovitushjelman-sa. Tällaisia ovat pienen kontaktorin, edellisen apukyt-kimen, riviliittimien ja lämpöreleen asennus. Esimerkkinä näistä on kontaktorin apukyt-kimen asentava ohjelma PLAHJEL3 (liite C, sivu 77). Apukyt-kin asennetaan pienen kontaktorin päällä olevaan uraan. Pieni kontaktori puolestaan kiinnit-tyy salvan avulla asennuskiskoon, joka on ruuvattu alustaan. Sekä ruuvauksessa että salpakiinnityksessä voi esiintyä epätarkkuuksia. Tämän vuoksi apukyt-kimen asennusta on val-vottava sensoreiden avulla. Asennusta valvotaan kouran pu-ristusta tuntevan voimasensorin avulla.

PLAHJEL3 jakautuu kahteen vaiheeseen, joista ensimmäinen varmistaa sen, että komponentti on oikealla asennusuralla kontaktorin päällä. Mikäli kouran voimasensori havaitsee liian suuren vastustavan voiman asennuksen alkuvaiheessa, keskeytyy asennus. PLAHJEL3 poistaa apukyt-kimen kokoonpa-

noalueelta ja antaa virheilmoituksen käyttäjälle. Muussa tapauksessa robotti irroittaa otteensa, jolloin apukytin asettuu asennusuraansa oikein. Tämän jälkeen robotti työntää komponenttia urassaan niin pitkälle, että lukittuminen salvan avulla tapahtuu. Lukittuminen havaitaan asennusvoiman kasvamisena yli annetun raja-arvon. Lopuksi robotti poistuu apukytimen yläpuolelle.

7.3.3.3 Komponenttien ruuvaukset

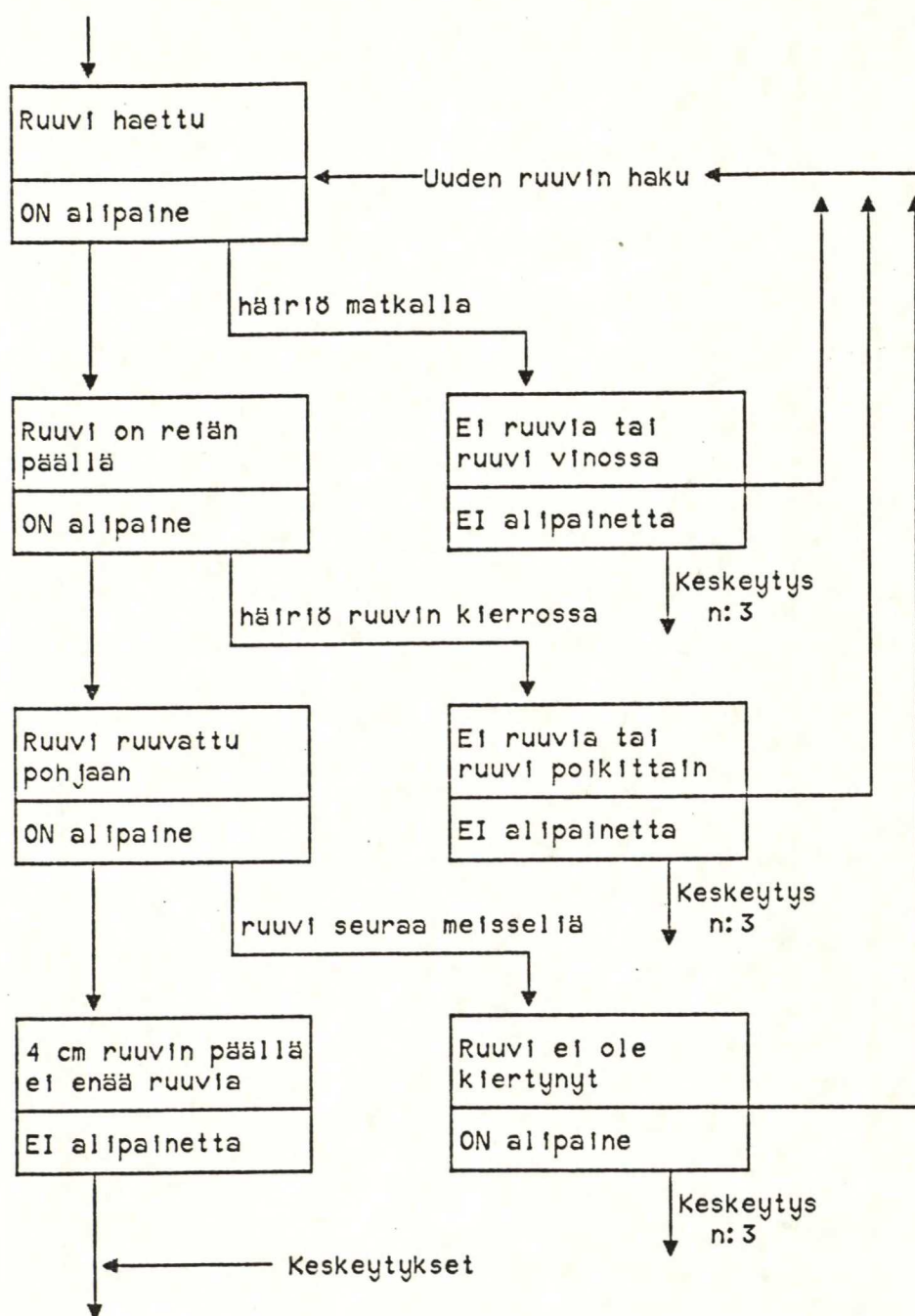
Ruuvausaliohjelmaa SKRV käytetään kaikkien ruuvien ruuvaamiseen (liite C, sivut 82-84). Ohjelma hakee uuden ruuvin ruuvitelimestä, vie sen halutun reiän yläpuolelle, ruuvaa ruuvin ja lopuksi vie ruuvitaltan pois työalueelta. Tätä toimintasarjaa voivat häiritä lukuisat eri virhetilanteet, jotka aiheutuvat ruuvitaltan, ruuvien ja ruuvinreikien valmistusepätarkkuuksista. Ruuvaukseen ei saa jäädä havaitsemattomia virheitä, sillä ne peittyvät asennettavien komponenttien alle. Ruuvit ovat pitkiä, ja melko herkkiä menemään vinoon. Robotti havaitsee virhetilat ruuvi-imun alipaineen valvonnan avulla. Jokaisessa työvaiheessa virhetila ilmenee siitä, että ruuvi ei seuraa tai seuraa virheellisesti ruuvitalttaa. Havaittuaan virheen robotti vie vanhan, mahdollisesti viallisen ruuvin hylkäyspaikalle ja hakee uuden ruuvin ruuvattavaksi. Mikäli ruuvaus ei onnistu kolmen yrityksen jälkeen, siirtyy robotti ruuvaamaan seuraavaa ruuvinreikää. Kaikista häiriöistä ilmoitetaan käyttäjälle kirjoittimen avulla.

Jotta ruuvausohjelma suoriutuisi kaikista ruuveista, sekä pitkistä että lyhyistä, ja kaikista ruuvinrei'istä, tarvitsee se seuraavia parametritietoja:

- Ruuvin reiän X-koordinaatti
- Ruuvin reiän Y-koordinaatti
- Ruuvattavan kappaleen korkeus ruuvin reiän kohdalta
- Turvallinen lähestymissuunta reiän päälle

Haettavan ruuvin koordinaatit saadaan pitämällä kirjaa kulu-tetuista ruuveista.

Ruuvausohjelma SKRV toteuttaa seuraavan tilakaavion, josta ilmenevät virhetilat ja korjaustoimet (Kuva 21 sivulla 58).



Kuva 21. SKRV-ohjelman yksinkertaistettu tilakaavio

Vahinkojen syntyminen esim. silloin, kun ruuvi ei lähde kiertymään estetään valvomalla ruuvausvoimaa. Sallitun voiman ylittyessä keskeytyy ruuvaus. Mikäli ruuvi ei ole kiertynyt kiinni havaitsee ohjelma sen ja korjaa tilanteen.

Testausajot ovat osoittaneet, että SKRV-ohjelma selviytyy kaikista normaalisti sattuvista virhetilanteista siten, että kokoonpano ei keskeydy. Ohjelma lisäksi korjaa virheen suurimmassa osassa virhetilanteita. Kaikista virheistä tulostuu aina virheilmoitus kirjoittimelle, joten käyttäjä voi myöhemmin tarkistaa virheelliset ruuvaukset.

7.3.4 Käyttökokemuksia

7.3.4.1 Ohjelmankehitys

Tässä sovelluksessa käytettiin aikaisemmissa kokeissa käytettyjä työkaluja, jotka eivät täysin soveltuneet sähkökomponenttien käsittelyyn. Tästä seurasi virhetilanteiden kasvu, mikä hallittiin kehittyneellä ohjelmoinnilla ja sensoreiden käytöllä. AML-ohjelmointikieli mahdollisti virheen korjauksen kesken ohjelman suoritusta ja ohjelman jatkamisen edelleen. Näin säästyi huomattavasti aikaa ja korjausmenetelmiä voitiin testata välittömästi kokoonpanon ollessa oikeassa vaiheessa. Näin vältettiin virhetilanteiden kehittyminen sellaiseksi, että toiminnan normaali jatkaminen olisi mahdotonta.

Joustava tietojenkäsittely-ympäristö ja valmiit oheislaitteet mahdollistivat sovelluksen kehittämisen noin kuudessa viikossa. Tuotantosovelluksen kehittäminen vaatii lisäksi arviolta kahden viikon työn. Tämä arviolta kahdeksan viikon kehitysaika on kilpailukykyinen muiden toteutusvaihtoehtojen kanssa.

7.3.4.2 Kalustusohjelman toiminta

Testiajojen ja käyttökokemusten perusteella voi ohjelmiston rakennetta ja toimintaa pitää onnistuneena. Tapahtuneiden kokoonpanovirheiden esiintymistiheys vaihteli suuresti työkalujen kunnon ja ruuvien laadun mukaan. Keskimäärin yksi ruuvausvirhe sattui kunkin kokoonpanon aikana. Ohjelma havaitsi virheen aina ja kykeni korjaamaan sen, mikäli virhe aiheutui ruuvien väärästä asennosta. On tärkeää, että käyttäjä saa tiedon tapahtuneesta, jolloin jälkitarkastus voidaan kohdistaa tehokkaasti robotin ilmoittamiin virheellisiin työkappaleisiin.

Suorituskyvyn optimointiin voidaan valmiissa ohjelmassa käyttää nopeuksien valintaa. Käytettäessä siirtoliikkeille 80%:sta nopeutta ja sovitustiikkeille 5%:sta nopeutta ei toimintanopeus aiheuta virhetilanteita. Kokoonpanoaika

näillä nopeuksilla on noin 10 min. Robotin käyttämän ajan lyhentäminen on mahdollista poistamalla turhia välipysäyksiä, ja sijoittamalla komponentit työalueelle tarkoituksenmukaisemmin. Arvioitu minimiaika kokoonpanolle on noin 7 min. Samaan työhön käytetty aika manuaalisessa kokoonpanossa on arviolta 30 min, joten robotti vastaa noin viiden kokoonpanotyöntekijän työpanosta.

Koeohjelma osoitti, että virhetarkastelut suoritettiin riittävän tiheästi ja oikeilla menetelmillä, sillä robotti kykeni itsenäisesti käsittelemään kaikki tapahtuneet virhetilanteet.

7.4 JÄRJESTELMÄN KEHITYSTARPEET

Kokoonpanotestiohjelma sisältää vain välttämättömimmät toiminnot. Testien aikana oli tiettyä hystereesiä havaittavissa työkalujen, erityisesti ruuvitaltan asennossa. Epäonnistuneen ruuvauksen jälkeen ruuvitaltan pitkä varsi jäi hiukan vinoon ja aiheutti lisää ongelmia. Tämän vuoksi olisi ainakin x-y-z-kalibrointi paikallaan esim. aina epäonnistuneen ruuvauksen jälkeen. Kalibrointi vaatii tosin ulkopuolisten sensoreiden käyttöä, sillä ruuvitaltan monet jouset heikentävät robotin omien tuntosensorien tarkuutta niin, ettei niitä voida käyttää kalibrointiin. Toinen vaihtoehto on vaihtaa ruuvitaltta tukevampaan.

Kohdekoordinaatistojen käyttö tekisi sovelluksen ylläpidon joustavaksi, mutta välttämätöntä se ei ole. Sen sijaan jatkuvassa tuotannossa tarvitaan lisätoimintoja, jotka mahdollistavat miehittämättömän toiminnan. Tärkein lisäys ohjelmistoon on materiaalin siirron ratkaiseminen. Se edellyttää ainakin pallettien vaihtoa ja mahdollisesti kommunikointia muiden laitteiden kanssa. Lisäksi virhetilatastuksia on lisättävä, sillä robotin toiminta on saatava täysin turvalliseksi ja varmaksi.

Alustan jousitapit voivat olla haitaksikin silloin, kun ruuvin kärki jostain syystä luiskahtaa tapin sivuun. Tällöin tappi estää ruuvin menemisen reikään ja ruuvausyritys epäonnistuu varmasti. Ruuvitaltan rakennetta tulee muuttaa sellaiseksi, että se keskittää ruuvin kierteittensä eikä kantansa mukaan. Myös pyörimissuunnan vaihto ohjelmallisesti olisi hyödyllinen ominaisuus, jonka avulla voisi vääriin kierteille menneen ruuvin vielä irroittaa. Nämä kaikki mekaaniset ongelmat voidaan ratkaista muuttamalla työkalujen ominaisuuksia, joten niistä ei odoteta aiheutuvan vaikeuksia tuotantosovelluksessa.

7.5 ROBOTISOINNIN VAIKUTUS ORGANISAATIOON

7.5.1 Vaikutus tuotesuunnitteluun

Sarjakoon kasvattamiseksi voidaan tuotevariantteja vähentää standardoimalla kytkinkaappien kalustusta. Vaikka kytkinkaapit on tilattu aiemmin yksilöinä, on niissä käytännössä vain niin pieniä eroja, ettei tuotemuutoksia tarvitse kovin paljoa tehdä. Suunnittelukonttorin kalustuspiirustusten tuottaminen vähenee huomattavasti. Vapautunut suunnitteluresurssi voidaan käyttää robottiohjelmien suunnitteluun. Näin myös suunnittelun on otettava huomioon robotin tarjoamat mahdollisuudet ja toisaalta sen toiminnan rajoittuneisuudet.

Osa komponenteista on teetetty kotimaassa alihankkijoilla. Näihin saadaan helposti muutokset, jotka tekevät kappaleeseen tarttumisen helpoksi, ja mahdollistavat ruuvitaltan pääsyn ruuvireikien päälle. Muutoksia ulkomaisiin standardikomponentteihin ei ole yhtä helppoa saada, mutta robottikokoonpanon yleistyessä myös näihin on odotettavissa muutoksia.

7.5.2 Vaikutus työhön

Robotin asennus ympäristöön, jossa aikaisemmin on tehty työtä täysin manuaalisesti, voi aiheuttaa ongelmia työmotivaatioon. Robotti vaatii aluksi jatkuvaa valvontaa ja virheiden raportointia ja korjailua. Työntekijöiden vastuun kasvaessa olisi parasta, jos nykyisestä urakkatyöstä päästäisiin eroon, jolloin robottia ei enää koettaisi kilpailijana ja urakoiden pilaaajana.

Robotti ei tule tekemään kaikkia työvaiheita. Jäljelle jää alustapeltien valmistus ja käynnistinkaapin johdotus. Kuljetuspallettien lataaminen ja purkaminen tuo mukanaan uusia työtehtäviä. Valmistuksen laadun nostaminen ja toiminnan tuottavuuden saattaminen kilpailukykyiselle tasolle tulee antamaan uusia työtehtäviä ja lisäämään vanhojen tehtävien mielekkyyttä. Etenemällä automaatiossa vaiheittain ja kouluttamalla henkilökunta tarkoituksenmukaisesti taataan työntekijöiden sopeutuminen muuttuviin työtehtäviin.

7.5.3 Koulutus

Koska robotti on joustava laite, ja uuden sovelluksen ottaminen käyttöön on vaivatonta, seuraa tästä järjestelmän jatkuva kehittämistarve. Tämän vuoksi on ainakin yhden henkilön on hallittava AML-ohjelmointikieli ja robotin toiminnan teoria. Myös operointiin normaalin toiminnan aikana tarvitaan koulutuksen saaneita henkilöitä. IBM:n Järfällan tehtaalla on robotin läheisyyteen sijoitettu manuaalisia työasemia, joiden työntekijät valvovat ja operoivat robottia oman työnsä ohessa. Näiden henkilöiden koulutus voidaan antaa paikan päällä.

8.0 TYÖN ARVIOINTI

Tässä diplomityössä olen kuvannut kokoonpanotöiden robotisointiin liittyviä erityiskysymyksiä sekä esitellyt erään kokoonpanotyöhön soveltuvan robotin rakenteen ja ominaisuuksia.

Ympäristön havainnointi on kokoonpanotehtävissä keskeistä, sillä virhetilanteiden mahdollisuus on aina olemassa. Virhetilanteita helposti seuraavien törmäysten seuraukset ovat niin haitallisia, ettei niitä voida tuotantosovelluksissa sallia. Ainoa keino virhetilanteiden hallitsemiseen teollisuusympäristössä on sensorien käyttö.

Tämän diplomityön osatavoitteena oli kytkinkaapin kokoonpano robotilla. Robotin tuli toipua useimmista käytännössä esiintyvistä virheistä. Sovelluksessa käytettiin robotin kouran tuntosensoreita sekä ulkopuolista painekeyhtä ympäristön havainnointiin. Näiden sensoreiden avulla saatiin tieto virheistä, mikä mahdollisti korjaustoimet.

Kokeen tuloksena oli onnistunut kokoonpano, joka toipui kaikista loppukokeessa esiintyneistä virhetilanteista. Kokoonpanoaika oli noin neljännes manuaalisesti työhön kulutetusta ajasta. Kokoonpanoaikaa voidaan lyhentää työliikkeiden huolellisella suunnittelulla vielä arviolta 30 %, mutta merkittävä nopeuden lisä saavutetaan vasta , kun työkohteesta saadaan lisää tietoa esim. kehittyneen näköaistin avulla.

Koe osoitti, että myös suhteellisen epätarkasti valmistettuja komponentteja voidaan käyttää robottikokoonpanossa. Tämä vastaa teollisuutemme tarpeita ja todennäköisesti edesauttaa robottien käytön yleistymistä myös kokoonpanotehtävissä.

9.0 KIRJALLISUUTTA

- /1/ J.R.Bailey, 'Product Design for Robotic Assembly', 13th ISIR and ROBOTS 7, Conference Proceedings, April 17-21, s.11-44, 1983
- /2/ Jeanine Meyer, 'An Emulation System for Programmable Sensory Robots', IBM J.Res.Develop., vol 25, No 6, s.955, 1981
- /3/ J.Y.S.Luh, 'An Anatomy of Industrial Robots and Their Controls', IEEE Transactions on Automatic Control, vol AC-28, No 2, s.133, 1983
- /4/ P.B.Scott, T.M.Husband, 'Robotic Assembly: Design, Analysis, and Economic Evaluation', 13th ISIR and ROBOTS 7, Conference Proceedings, April 17-21, s.5-12, 1983
- /5/ L.I.Lieberman, M.A.Wesley, 'AUTOPASS: An Automatic Programming System for Computer Controlled Mechanical Assembly', IBM J.Res.Develop., vol. 21, s.321, 1977
- /6/ M.Sheier, E.Kent, J.Albus, P.Mansbach, M.Nashman, L.Palombo, W.Rutkowski, T.Wheatley, 'Robot Sensing for a Hierarchical Control System', 13th ISIR and ROBOTS 7, Conference Proceedings, April 17-21, s.14-50, 1983
- /7/ IBM 7535 Manufacturing System User's Guide, IBM 1982
- /8/ M.A.Lavin, L.I.Lieberman, 'AML/V: An Industrial Machine Vision Programming System', The International Journal of Robotics Research, Vol 1, No 3, s.42, 1982
- /9/ R.H.Taylor, D.D.Grossman, 'An Integrated Robot System Architecture', Proceedings of the IEEE, Vol 71, No 7, s.842, July 1983
- /10/ R.H.Taylor, P.D.Summers, J.M.Meyer, 'AML: A Manufacturing Language', Research Report RC-9289, IBM T.J.Watson Research Center Yorktown Heights, NY, 1982
- /11/ Antti J. Niemi, 'Control of robots and manipulators by automatic visual observation', Sähkö, vol. 57, No 2, s.10, 1984
- /12/ Tomas Lozano-Perez, 'Robot Programming', Proceedings of the IEEE, vol. 71, No. 7, s.821, July 1983
- /13/ IBM Robot System/1 Maintenance Information Manual, IBM 1982
- /14/ A Manufacturing Language Reference, IBM 7565 Manufacturing System Software Library, IBM 1982

/15/ M.T.Mason, 'Compliance and Force Control for Computer Controlled Manipulators', IEEE Transactions on Systems, Man, and Cybernetics, vol SMC-11, No 6, s.418, 1981

/16/ D.Haaf, 'An Industrial Robot Using Tactile and Visual Sensory Feedback for Assembly Process Control', Proceedins of the 5th International Conference on Automated Inspection and Product Control, Stuttgart, Germany, s.165, 24-26 June 1980

/17/ A.C.Sanderson, G Perry, 'Sensor Based Assembly Systems: Research and Applications in Electronic Manufacturing', Proceedings of the IEEE, vol. 71, No 7, s.856, July 1983

/18/ J.S.Albus, A.J.Barbera, M.L.Fitzgerald, 'Programming a Hierarchical Robot Control System', Proc. 12th International Symposium on Industrial Robots, Paris, France, 9-11 June 1982

/19/ 'Sensors the key to robot of the future', Conference report, Industrial Robot, vol 10, No 4, Dec 1983

LIITE A. SIGMA-KOKOONPANOASEMA

Olivetti:n SIGMA-robotista on kehitetty teollisuussympäristöön soveltuva kokoonpanoasema, joka on varustettu sensoreilla ja kehittyneellä ohjausjärjestelmällä /16/. Robotti käyttää kosketusaistia ja näköhavaintoja ohjaamaan kokoonpanoa. Aistit on kuitenkin toteutettu siten, ettei robotin hinta kasva liian suureksi. Nopeus, luotettavuus ja käytön helppous olivat myös tärkeitä suunnittelukriteereitä.

Sensoreiden käyttö: 1. Vertaile sensori-informaatiota ennalta ohjelmoituun malliin (tosi/epätosi); 2. Mittaa sensorista saatavasta tiedosta ja käytä tietoa korjaustoimen ohjaamiseen. Usein valvotaan jonkin rajaehdon ylitystä.

Tässä esimerkissä pyritään informaation minimointiin, sillä sensoreiden erottelukyky ja ohjausyksikön laskentakapasiteetti rajoittavat suuren tietomäärän nopeaa käsittelyä.

Sensorien prosessointi on täysin integroitu ohjausjärjestelmään (Kuva 22 sivulla 68). Sensorit ovat myös fyysisesti olennainen osa robotin rakennetta. Tämä rajoittaa voimakkaasti esim. kosketusanturin kokoa ja painoa.

Laitteisto koostuu porttaalirakenteisesta SIGMA-robotista, johon on lisätty ominaisuuksia, jotka mahdollistavat suhteellisten pienten sarjojen kokoonpanon. Eri sarjojen yhteenlaskettu tuotanto riittää työllistämään robotin kannattavalla tavalla.

Robotin komponentteja ovat:

- Ranne, joka kykenee vaihtamaan tarttujaansa ja työkaluja automaattisesti.
- Kosketusanturi, joka on varustettu kuuden akselin mukautuvaisuuselimellä ja yhden akselin voima- tai reitinsuranta-anturilla.
- Nopea ja halpa TV-kamera visuaalisiin tarkastuksiin. Kameraa ohjataan robotin SIGLA-ohjelmointikielestä suoraan.
- Vaihdeettava työskentelyalusta, jolla aikaansaadaan työkohteen nopea vaihto.

Kosketusanturiin on lisätty mekaanisen mukautumiselimien lisäksi asemanmittaus, jolla voidaan tutkia vallitsevaa asennusvoimaa tai käyttää sitä radanmittaukseen. Mittaus suoritetaan ainoastaan yhden akselin suuntaan, jolloin tarvittavat laskutoimitukset ovat hyvin yksinkertaisia. Koska kosketuslaitteisto on suhteellisen monimutkainen kokonaisuus, on se integroitu robotin ranteeseen eikä se ole vaihdettava. Mukautumiselin ei yksinään selviydy kaikista sovitustehtävistä, vaan se tarvitsee tuekseen robotin avus-

tavia liikkeitä. Mikäli kappaleitten väliset liittämisoimat tulevat liian suuriksi, on robotin mahdollista reagoida tähän ja suorittaa tarvittavat apuliikkeet. Robotilla voidaan etsiä oikea asennusorientaatio tuntoelimen avulla ja näin saada aikaan luotettava kokoonpanon kulku. Asennus voi tyydyttävästi kestää kolme sekuntia, mutta aika voi kasvaa 10-30 %, mikäli tarvitaan apurutiineja.

Tarttujassa on sisäinen keskitystarttuja, jonka tehtävänä on keskittää mukautumiselin, mikäli mukautumista ei tarvita. Esimerkiksi jos käsitellään pitkiä kappaleita, jotka eivät ole pystysuorassa, tulee mukautumiselimen toiminta epämääräiseksi, koska se ei erota kappaleeseen ennen asennusta kohdistuvaa vääntömomenttia asennuksen aikana syntyvistä vastustavista voimista.

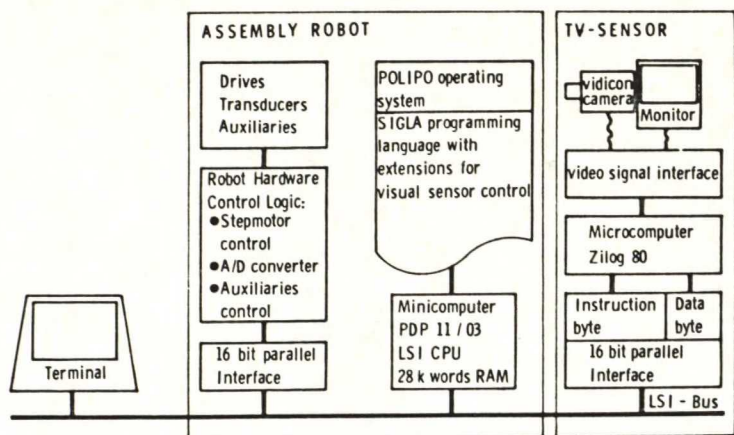
Visuaaliset tarkastustehtävät työllistävät lukuisia ihmisiä tehtaissa. Se on pieni, mutta olennainen osa valmistusprosessia. SIGMA-robotin yhteyteen rakennetun näköaistin tehtävänä on tutkia komponenttien ja työkalujen olemassaoloa ja oikeaa orientaatiota syöttölaitteissa robotin kourassa. Tavoitteena oli aikaansaada mahdollisimman yksinkertainen, halpa, nopea ja helposti ohjelmoitava näkötoiminta, joka on integroitu robotin ohjausjärjestelmään.

Näköaisti on hyvin yksinkertainen. Se perustuu kahden kuva-alueelta valittavissa olevan havainnointijanan alueella esiintyvien musta-valko-rajakohtien etäisyyden tutkimiseen. Eri rajakohtakuvioita voidaan talletta muistiin ja toimintavaiheessa verrataan opetettua rajakuviota kameralla havaittuun. Koska muistiin talletetaan ainoastaan kuvion muoto, eikä sen sijaintia kuva-alueella, voidaan havainnointijanojen sijaintia vaihtelemalla saada selville osan sijainti ja orientaatio. Tästä yksinkertaisuudesta seuraa hyvin lyhyt vasteaika - alle 40 ms - , ja luotettava toiminta. Näköjärjestelmän liittäminen robotin ohjelmointikieleen on tuonut mukanaan viisi lisäkäskeyä. Optinen havainnointitarkkuus on noin yksi millimetri. Näköaistilla suoritettavia toimintoja ovat:

- Robotin tarttujassa olevan kappaleen olemassaolo, orientaatio, muodon tarkistus.
- Robotin tarttujan tai työkalun aseman tarkistus.
- Kokoonpanotyön editymisen valvonta.
- Asennuksen ohjaus yhdessä tuntoaistin kanssa. Esimerkiksi asennettavan tapin kaatuminen ja kaatumisen suuruus.

TV-kameran toiminta on riittävän nopeata ohjaamaan pysäyttämään robotin liike haluttuun paikkaan.

Robottilaitteiston osien kehittäminen teollisuusympäristöön soveltuvaksi on työn alla.



Kuva 22. Sigma-kokoonpanoaseman ohjauslaitteisto

LIITE B. PIIRILEVYN KOKOAMISASEMA

Esimerkkinä laboratoriossa kehitetty järjestelmä SEAS (Standard Electronic Assembly Station), jonka on kehittänyt Westinghouse Electric Corporation piensarjatuotantoon /17/.

Kokoonpanolaitteiston tehtävänä on asentaa erilaisia komponentteja piirilevylle. Järjestelmän tulee kyetä käsittelemään suorakaiteen muotoisten komponenttien ohella myös aksiaalikomponentteja, joiden liitântäjohtimet eivät ole suunniteltu automaattiseen kokoonpanoon. Järjestelmän työvaiheisiin kuuluu komponentin asettaminen paikalleen, juottaminen ja johtimien katkaisu määrämittään, sekä valmiin tuotteen testaus ja siirto varastoon.

Kokoonpanojärjestelmän tulee voida käsitellä erityyppisiä ja -muotoisia komponentteja. Komponenttien pakkaustiheys kortille on suuri; ne voivat olla kosketuksessa toistensa kanssa. Myös piirilevyt vaihtelevat so. järjestelmän tulee kyetä valmistamaan useita eri tuotteita. Sarjakoko on hyvin pieni. Pyrkimyksenä on päästä jopa vain 6 - 10 korttia käsittävien sarjojen joustavaan tuotantoon. Tuotannossa ja tuotteissa voi tapahtua usein muutoksia. Laatu- ja luotettavuusvaatimukset ovat erittäin korkeat.

Kokoonpanolaitteisto muodostuu eri yksiköistä seuraavasti: Piirilevyn yläpuolella on manipulaattori, joka on varustettu kouralla, johon on asennettu neljä erillisesti liikuteltavaa sormea. Tämän manipulaattorin tehtävänä on tuoda asennettavan komponentin käsiteltäväksi oleva johdin piirilevyssä olevan reiän välittömään läheisyyteen. Kortin alapuolelle on asennettu X-Y-pöydän avulla liikkuva videokamera, joka havaitsee asennusreiät sekä laskee niiden keskipisteet. Kamera havaitsee myös johtimen pään erityisvalaistuksen avulla ja kykenee näin ohjaamaan manipulaattoria siten, että reikä ja johdin ovat vastakkain. Kortin alapuolella sijaitsevat lisäksi johtimien päiden katkaisulaitteet sekä juottamiseen tarvittavat laitteet.

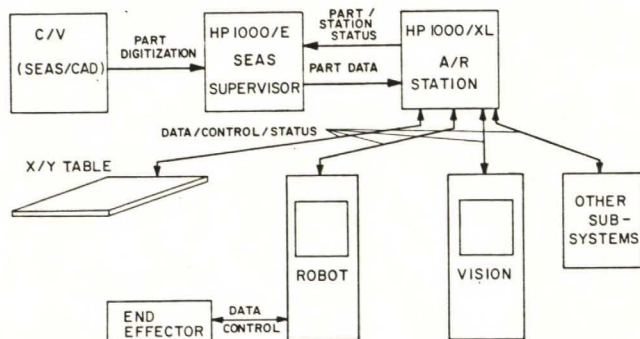
Sensoreista mainittiin jo videokamera. Järjestelmässä on käytetty myös rajakytkimiä indikoimaan osan olemassaoloa. Robotin kouraan on kytketty voima-anturit. Niiden tehtävänä on tarttumisen kontrollointi ja komponenttien toleranssien kompensointi. Asennusalustaan on kiinnitetty x- ja y-suunnan voimia aistivat anturit, joiden avulla lopullista komponentin asennusta voidaan valvoa ja tarvittaessa suorittaa tarpeelliset virheidenkorjausrutiinit.

Ohjausjärjestelmä on hierarkkinen (Kuva 23 sivulla 70). Alimmalla tasolla ovat järjestelmän komponenttien ohjausyksiköt kuten X-Y-pöytä, robotti, näköjärjestelmä ja muut oheislaitteet. Näitä järjestelmiä ohjaa HP 1000/XL-tietokone, joka puolestaan kommunikoi kokoonpanoasema ohjaavan HP 1000/E-tietokoneen kanssa lähettäen tälle

tietoja kokoonpanon tilasta ja saaden komponenttien tietoja. HP 1000/E on yhteydessä CAD-järjestelmään.

Kappaleiden tiedot on esitetty yhteisessä tietokannassa, joka saadaan CAD-järjestelmän avulla. Kappaleiden geometrisiin tietoihin on lisätty kokoonpanomenetelmään liittyviä tyyppitietoja. Näin uusien komponenttien kokoonpano-ohjeet voidaan antaa järjestelmälle muuttamalla ainoastaan tietokannan sisältöä.

Laitteisto on kehitetty erillisessä laboratoriossa, jossa myös tuotantokäytön vaatimukset on otettu huomioon. Asema oli artikkelin kirjoitushetkellä tarkoitus siirtää Westinghouse-yhtiön tuotantolaitoksille joustavaan tuotantoon kokoonpanotehtäviin. Näin ollen käyttökokemuksia ei ole saatavilla.



Kuva 23. SEAS-aseman ohjauslaitteisto

LIITE C. OHJELMALISTAUKSET

```

10: --FILE: 'WDEMO.USER01'
20: -----*
30: --*
40: --*
50: --*
60: --*
70: --*          WDEMO
80: --*
90: --*          UTPRINTAD DEN/ LISTATTU/ ..19../.5... 1984
100: --*
110: --*
120: -----*
130: --
140: -- PAIVITETTY: 16.8.1984 JAAKKO PERTTU
150: --
160: -- OHJELMA KAYTTAA SEURAAVIA MENUTIEDOSTOJA:
170: --
180: -- MENLANG.          KIELIVALINTA
190: -- MAPPLSWE.         SOVELLUKSEN VALINTA
200: -- MAPPLFIN.         -- ;; --
210: -- MCLSWE.           KELLON ASETUS
220: -- MCLFIN.           -- ;; --
230: -- MINSSWE.          TUOTE A:N OHJEET
240: -- MINSFIN.          -- ;; --
250: --
260: -----
270: -- LADATAAN TARVITTAVAT TIEDOSTOT
280: -----
290: IF ?SHOW EQ 0 THEN LOAD('SHOW.SYS01');
300: DISPLAY('NU LADDAS WDEMO',EOL);
310: -----
320: -- MAARITTELYT
330: -----
340: FIN: STATIC $1;
350: SWE: STATIC $0;
360: LANG: NEW FIN;
370:
380: APPL: NEW INT; -- 0.....END
390: -- 1.....SET THE SYSTEM CLOCK
400: -- 2.....7565 DEMO
410:
420: DEMO: SUBR;
430: --
440: -- PAAOHJELMA, JOKA SISALTAA MENU-OHJATUN
450: -- KAYTTAJALIITANNAN
460: -- WDEMON SOVELLUSTEN KAYNNISTYS.
470: -- ENSIN KYSYTAAN OHJEIDEN KIELI RUOTSI/SUOMI
480: --
490: LANG=SHOW('MENLANG',FIN); -- VALITAAN OHJEIDEN KIELI
500: REPEAT
510: BEGIN
520: IF LANG EQ SWE
530: THEN APPL=SHOW('MAPPLSWE')
540: ELSE APPL=SHOW('MAPPLFIN');
550: IF APPL EQ 1 THEN CLOCK1
560: ELSE IF APPL EQ 2 THEN ROBDEN;
570: END
580: UNTIL (APPL EQ 0);
590: END; --DEMONAIN

```



```

610: -----
620: HEADER: SUBR;
630:   DISPLAY(EOP,SYSS(14),
640:           '*****',EOL);
650:   DISPLAY(SYSS(14),
660:           '***' ,EOL);
670:   DISPLAY(SYSS(14),
680:           '***   WARTSILA   ROBOTICS   ' ,EOL);
690:   DISPLAY(SYSS(14),
700:           '***' ,EOL);
710:   DISPLAY(SYSS(14),
720:           '*****',EOL);
730: END;  --HEADER
740: -----
750: SYSS: SUBR(N);  -- SHORT CALL TO STRING (N)
760:   RETURN (STRING(N));
770: END;  --SYSS
780: -----
790: CLOCK1: SUBR;
800: --
810: -- SYSTEEMIKELLON ASETUS
820: --
830: TIME1: NEW (0,0,0,0,0,0,0);
840: TIME2: NEW (0,0,0,0,0,0,0);
850:
860: TIME1=TIME;
870:
880: IF LANG EQ SWE
890: THEN TIME2=SHOW('MCLSWE',TIME1)
900: ELSE TIME2=SHOW('MCLFIN',TIME1);
910:
920: IF TIME2 NE TIME1
930: THEN TIME(TIME2(1),TIME2(2),TIME2(3),TIME2(4),TIME2(5),
940:           TIME2(6),TIME2(7));
950:
960: END;  -- CLOCK1
970: -----
980: RMSG: STATIC STRING(100);
990: TIME3: NEW (0,0,0,0,0,0,0);
1000: RAPO: SUBR(RMSG);
1010: --
1020: -- ALIOHJELMA LISAA KELLONAJAN SAATUUN VIESTIIN
1030: -- JA TULOSTAA RIVIN KIRJOITTIMELLA.
1040: --
1050: TIME3=TIME;
1060: PRINT(1,(TIME3(1),'.',TIME3(2),'.',TIME3(3)));
1061: PRINT(1,(SYSS(9-COLUMN(1)), ' ==> '));
1070: PRINT(1,(RMSG,EOL));
1080: END;  --RAPO

```

```

1090: -----
1100: ROBDEN: SUBR?
1110: --
1120: -- TAMA ALIOHJELMA SUORITTA A KAIKKI TUOTTEEN A KOKOONPANOON
1130: -- LIITTYVAT TOIMET, KUTEN OHJEIDEN ANTAMISEN, TARPEELLISET
1140: -- KYSELYT, ITSE TYON SUORITUKSEN JA RAPORTOINNIN.
1150: --
1160: IF LANG EQ SWE THEN SHOW('MINSSWE') ELSE SHOW('MINSFIN');
1170: DISPLAY(EOP);
1180: DISPLAY('
1181: DISPLAY('
1182: DISPLAY('      |  IBM 7565/RS 1
1183: DISPLAY('      |
1184: DISPLAY('      +-----+
1185: DISPLAY('      ||          ||          |  |
1186: DISPLAY('      ||          ||          |  |
1190: DISPLAY('      ||          ||          |  |
1200: DISPLAY('      ||          ~~~          |  |
1210: DISPLAY('      ||          |          |  |
1220: DISPLAY('      ||          |          |  |
1230: DISPLAY('      ||-----+-----+  |
1240: DISPLAY('      ||-----+-----+  |
1250: DISPLAY('      ||          ||          |  |
1260: DISPLAY('      ||          ||          |  |
1270: DISPLAY('      ||          ||          |  |
1271: DISPLAY('      ||          ||          |  |
1272: DISPLAY('
1273: DISPLAY('
1274: DISPLAY('      NYT ALOITAN TYOSKENTELYN !      ',EOL);
1275: DISPLAY('
1276:
1277: DELAY(1); -- VAROAIKA
1278:
1281: SEQW; --KUTSUTAA SOVELLUSOHJELMAA.
1282:
1290: END; --ROBDEM

```

```

10: --FILE:'WART' UPPL 8/8-84 A.R. 30/8-84 J.P.
20: -----
30: --*
40: --*
50: --*
60: --*
70: --* WART
80: --*
90: --* UTPRINTAD DEN 12./1... 1984
100: --*
110: --*
120: -----
130: --
140: LOAD('NAR6');
150: DISPLAY('NU LADDAS WARTSILA',EOL);
160: IF ?WDEMO EQ 0 THEN LOAD('WDEMO');
170: IF ?SKRV EQ 0 THEN LOAD('WARTSTSK');
180: IF ?SYSGRIP EQ 0 THEN LOAD('SYSGRIP');
190: IF ?SYSGRAS EQ 0 THEN LOAD('SYSGRAS');
200: IF ?WRTDEF EQ 0 THEN LOAD('WRTDEF'); -- DEFIO-FILEN
210: IF ?PA EQ 0 THEN LOAD('RULDEF');
220: -----
230: H: STATIC .10; -- DEFAULT HASTIGHET AR .10
240: SAKT: STATIC .05; -- LANGSAMMA ROELSER AR .05
250: ZKORR: STATIC -28.0; --ALUSTAN LASKUN Z-KORJAUS
260: TRSPT: STATIC (-9,-150,140,0); --VALIPISTE X-Y-Z-PITCH
270: TRSPT2: STATIC (10,-313,215); --RUUVVAKSEN VALIPISTE
280: SPOTPLATS: STATIC (-1,-215,37); --HYLKAYSPAIIKKA
290: KOPX: STATIC 164.1; --TOKOPL POS 1. POS 2 = 129.7
300: -----
310:
320: SEQW:SUBR;
330: --
340: -- RESET MONITORS
350: --
360: ENDMONITOR((1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16));
370: --
380: WHILE SENSIO(LED,0) DO TARKISTA LED-KENNOJEN AVULLA, ETTA
390: BEGIN ROBOTIN KOURA ON VARMASTI TYHJA.
391: IF LANG EQ FIN THEN
392: BEGIN
400: DISPLAY('ROBOTIN KOURASSA ON JOTAKIN!',EOL);
410: BREAK('KORJAA, JA PAINA RETURN',EOL);
411: END
412: ELSE BEGIN
413: DISPLAY('DET FINNS NAGAT I GRIPPERN!',EOL);
414: BREAK('FIXA OCH TRYCK RETURN',EOL);
415: END;
420: END;
430: --
440: TID:NEW CLOCK;
450: PRINT(1,<EOL,EOL,EOL,EOL>);
451: IF LANG EQ FIN THEN
452: BEGIN
460: RAPO('KALUSTUS ALKAA');
470: PRINT(1,<SYSS(19),'NOPEUDET OVAT ',H,' / ',SAKT,EOL>);
471: END
472: ELSE BEGIN
473: RAPO('MONTERING BORJAR');
474: PRINT(1,<SYSS(19),'HASTIGHETERNA AR ',H,' / ',SAKT,EOL>);
475: END;
480: GO(3,200);

```



```

481:      --
490:      TAGRIP1;
500:      GO((1,2,3,5),TRSPT);  -- GA UNKAN
520:      TAKONTEL4;
530:      PLAKONTEL4;      ASETA VIRTAKYTKIN
550:      TAHALLARE;
560:      PLAHALLARE1;      ASETA SULAKEPESAT
570:      TAHALLARE;
580:      PLAHALLARE2;
590:      TAHALLARE;
600:      PLAHALLARE3;
620:      TATRAFO1;
630:      PLATRAFO1;      ASETA VIRTAMUUNTAJA
640:      LAGRIP1;
650:      --
660:      TAGRIP2;
680:      TAPLAT;
690:      FLAPLAT;      ASETA KYTKENTAKISKO
710:      TAKABRA;
720:      PLAKABRA;      ASETA KAAPELIKOURU
750:      LAGRIP2;
760:      --
770:      TASKRUVDR;
780:      GO(2,-260);  --VAPAA-ASEMA RUUVAA PIENET KOMPONENTIT
790:      SKRUVNING;  -- 3 X HALLARE, KONTEL4, TRAF01, PLAT
800:      GO((1,2,3),(188,-199,215));  -- VAPAA-ASEMA
810:      LASKRUVDR;
820:      --
830:      TAGRIP1;
840:      GO((1,2,3,5),TRSPT);  -- GA UNKAN
880:      TASPATR7;
890:      PLASPATR7;      ASETA JÄNNITEMUUNTAJA
910:      LAGRIP1;
920:      --
930:      BREAK('BYT GRIPPER I STATION 1 TILL DEN STORA, RETURN',EOL);
940:      --
950:      TAGRIP1;      -- STORA GRIPPERN PA SAMMA PLATS SOM GRIP1
960:      GO((1,2,3,5),TRSPT);  -- GA UNKAN
970:      TAKONTR8;
980:      PLAKONTR8;      ASETA SUURI KONTAKTORI
1000:     LAGRIP1;
1010:     --
1020:     BREAK('BYT GRIPPER I STATION 1 TILL DEN LILLA, RETURN',EOL);
1030:     --
1040:     TASKRUVDR;
1050:     GO(2,-260);      RUUVAA SUURET KOMPONENTIT
1060:     SKRUVNING2;  -- SPATR7, KONTR8
1070:     GO(1,195);
1080:     LASKRUVDR;
1090:     --
1091:     TAKONTR5;
1092:     PLAKONTR5;      ASENNA PIENI KONTAKTORI
1100:     TAHJEL3;      ASENNA APUKUTKIN KONPAKTORIN PÄALLE
1110:     PLAHJEL3;
1120:     --
1121:     -- TOISEN TYYPPISELLA KONTAKTORILLA TESTI
1122:     -- VOIDAAN TEHDÄ TUTKITAAN LÄMPÖRELEEN KIRISTYSRUUVIT
1130:     -- IF SKRUVTEST EQ FALSE THEN
1140:     --   RAPO('LÄMPÖRELEEN RUUVI(T) KIINNI, LÄMPÖRELETTÄ EI ASENNETA')
1150:     -- ELSE BEGIN
1160:     --   TATEREL6;
1170:     --   PLATEREL6;      ASENNA LÄMPÖRELE SUUREEN KONTAKTORIIN
1180:     -- END;
1190:     TAKOPL;
1200:     PLAKOPL;      RIVILIITTIMIEN (5 kp1) ASENNUS
1210:     RETKOPL;

```

```

1220:  --
1221:  IF LANG EQ FIN THEN
1222:  BEGIN
1230:      RAPO('ALUSTA KALUSTETTU');
1240:      PRINT(1,(SYSS(19),'KALUSTUS KESTI      ',CLOCK-TID,' S',EOL));
1241:  END
1242:  ELSE BEGIN
1243:      RAPO('UNDERLAGG MOBLARAD');
1244:      PRINT(1,(SYSS(19),'MOBLERING TOG      ',CLOCK-TID,' S',EOL));
1245:  END;
1250: END;
-----
1260:
1270: SKRUVTEST: SUBR;
1280:     UPPMON: NEW MONITOR(SLT,3,-10000,100);
1290:  --
1300:     SPEED(H);
1310:     GO(ARM,(-12.5, -625.2, 70.0, 44.0, -1.6, 0.4, 60.1));
1320:     SPEED(SAKT);
1330:     GO(3,43.0,UPPMON);      -- LAITIMMAINEN RUUVI
1340:     SPEED(H);
1350:     GO(3,70.0);
1360:     IF REMONITOR(UPPMON) THEN
1370:     BEGIN
1380:         GO((1,7),(0.1, 44.4));
1390:         SPEED(SAKT);
1400:         GO(3,43.0,UPPMON);      -- KESKIMMAINEN RUUVI
1410:         SPEED(H);
1420:         GO(3,70.0);
1430:         IF REMONITOR(UPPMON) THEN
1440:         BEGIN
1450:             GO((1,7),(9.2, 24.3));
1460:             SPEED(SAKT);
1470:             GO(3,43.0,UPPMON);      -- LAHIN RUUVI
1480:             SPEED(H);
1490:             GO(3,70.0);
1500:             IF REMONITOR(UPPMON) THEN
1510:             BEGIN
1520:                 ENDMONITOR(UPPMON);
1530:                 RETURN(TRUE);
1540:             END;
1550:             END;
1560:         END;
1570:         ENDMONITOR(UPPMON);
1580:         RETURN(FALSE);
1590:     END; -- SKRUVTEST
-----
1600:
1610: TAGRIP1: SUBR;
1620:  --      X      Y      Z      ROLL      PITCH YAW      GRIP0  GRIPS
1630:      SYSGRAS(-196.83,-301.65,-109.57, 45,-1.4446, 0, 36, 13);
1640:  END;
-----
1650:
1660: LAGRIP1: SUBR;
1670:  --      X      Y      Z      ROLL      PITCH YAW      GRIP0  GRIPS
1680:      SYSGRIP(-196.83,-301.65,-109.37, 45,-1.4446, 0, 13, 36);
1690:  END;
-----
1700:
1710: TAGRIP2: SUBR;
1720:  --      X      Y      Z      ROLL      PITCH YAW      GRIP0  GRIPS
1730:      SYSGRAS(-197.10,-501.65,-109.57, 45,-1.4446, 0, 36, 13);
1740:  END;
-----
1750:
1760: LAGRIP2: SUBR;
1770:  --      X      Y      Z      ROLL      PITCH YAW      GRIP0  GRIPS
1780:      SYSGRIP(-197.10,-501.65,-109.37, 45,-1.4446, 0, 13, 36);
1790:  END;
-----

```



```

1800: -----
1910: TATRAFD1:SUBR;
1920: --          X          Y          Z      ROLL  PITCH YAW  GRIPO  GRIPS
1930:      TOKO(29.20 ,690.55 , -91.57, 73.7,20.8621,  0,  36, 19);
1940: END;
1950: -----
1960: PLATRAFD1:SUBR;
1970: --          X          Y          Z      ROLL  PITCH YAW  GRIPO  GRIPS
1980:      PLAKO(-31.00,-475.38,51.00+ZKORR, 73.41,-.52468,.179,  19, 36);
1990: END;
2000: -----
2010: TAHALLARE:SUBR;
2020: --          X          Y          Z      ROLL  PITCH YAW  GRIPO  GRIPS
2030:      TOKO(44.77,633.57, -117.20, 92.99, 24.13, 0,  33, 17.5);
2040: END;
2050: -----
2060: PLAHALLARE1:SUBR;
2070: --          X          Y          Z      ROLL  PITCH YAW  GRIPO  GRIPS
2080:      PLAKO(129.68, -448.905, 31.00+ZKORR, 92.70, -2.49, 0, 18.5, 33);
2090: END;
2100: -----
2110: PLAHALLARE2:SUBR;
2120: --          X          Y          Z      ROLL  PITCH YAW  GRIPO  GRIPS
2130:      PLAKO(79.68, -448.905, 31.00+ZKORR, 92.70, -2.49, 0, 18.5, 33);
2140: END;
2150: -----
2160: PLAHALLARE3:SUBR;
2170: --          X          Y          Z      ROLL  PITCH YAW  GRIPO  GRIPS
2180:      PLAKO(29.68, -448.905, 31.00+ZKORR, 92.70, -2.49, 0, 18.5, 33);
2190: END;
2200: -----
2210: TAHJEL3:SUBR;
2220: --          X          Y          Z      ROLL  PITCH YAW  GRIPO  GRIPS
2230:      TOKO(18.65, 568.90,-164.49, 45.31 , 24.713, 0.04, 57, 42);
2240: END;
2250: -----
2260: PLAHJEL3:SUBR;
2270:      RMON: NEW MONITOR(SLP,3,-10000,-3000);
2280:      MMON: NEW MONITOR(SLP,1,-10000,2000);
2290:      --
2300:      GO((1,2,3,5),TRSP);
2310:      GO(ARM,(126.0, -609.8, 76.5+ZKORR, -44.1, -1.61, 0.04, 42.8));
2320:      REMONITOR(RMON);
2330:      GO(ARM,(127.83,-619.78, 61.51+ZKORR,-44.12,-1.61,0.04,42.8),RMON);
2340:      -- NYT APUKYTKIN ON EDESSA
2350:      SPEED(SAKT);
2360:      GO(2,-591.88,RMON);      ! LIIKE PYSÄHTYY, JOS ASENNUSVOIMA YLITTY
2370:      IF QMONITOR(RMON) THEN      (RMON= VANHA PURISTUSVOIMA-30N)
2380:      BEGIN
2390:          RAPO('APUKYTKIMEN ASENNUS EI ONNISTUNUT');
2400:          SPEED(H);
2410:          GO(2,-700);
2420:          GO(7,57);      POISTA KOMPONENTTI TYÖALUEELTA
2430:          END
2440:          ELSE BEGIN
2450:              SPEED(H);
2460:              GO(7,48);      -- OTETAAN PAREMPI OTE
2470:              GO(3,23.4);
2480:              GO(7,45.0);
2490:              SPEED(SAKT);
2500:              REMONITOR(MMON);
2510:              GO(2,-560,MMON);      TYÖNNETÄÄN, KUNNES LUKITTUU
2520:              SPEED(H);
2530:              GO((2,7),(-578,57));
2540:              END;
2550:              GO(3,125);
2560:              ENDMONITOR((RMON,MMON));
2570:      END;

```



```

2580: -----
2590: TAKONTEL4:SUBR;
2600: --      X      Y      Z      ROLL  PITCH YAW GRIPO GRIPS
2610:      TOKO(10.836, 480.45, -102.851, 80.72, -21.19, 0, 68, 44);
2620: END;
2630: -----
2640: PLAKONTEL4:SUBR;
2650: --      X      Y      Z      ROLL  PITCH YAW  GRIPO GRIPS
2660:      PLAKO(64.74, -604.33, 38.30+ZKORR, 79.56, -.8245, 0.0, 44, 68);
2670: END;
2680: -----
2690: TAKONTR5:SUBR;
2700: --      X      Y      Z      ROLL  PITCH YAW GRIPO GRIPS
2710:      TOKO(1.66, 393.93, -117.23, 45.19, 20.57, 0, 65, 54);
2720: END;
2730: -----
2740: PLAKONTR5:SUBR;
2750:      TRMON: NEW MONITOR(SLP,1,-10000,4500);
2760: --
2770:      GO(5,0);      -- RATA PITCHEN
2780:      GO((1,2,3,5),TRSPT);
2790:      GO((1,2,4,5,6),(-136, -620, 45.19, -7.80, -90.00 ));
2800:      GO((2,3),(-618.33, 0.39));
2810:      GO((2,7),(-612.0, 67));
2820: --
2830:      SPEED(SAKT);
2840:      REMONITOR(TRMON);
2850:      GO((2,5),(-540, 3),TRMON);
2860: --
2870:      SPEED(H);
2880:      GDLTA(JY,-5);
2881:      GO(3,150);
2882:      ENDMONITOR(TRMON);
2883: END; --PLAKONTR5
3070: -----
3080: TATEREL6:SUBR;
3090: --      X      Y      Z      ROLL  PITCH YAW GRIPO GRIPS
3100:      TOKO(-.97, 312.89, -105.71, 45.41, 17.99, 0.0, 79.0, 54);
3110: END;
3120: -----
3130: PLATEREL6:SUBR;
3140: --      X      Y      Z      ROLL  PITCH YAW  GRIPO GRIPS
3150:      GO((1,2,3,5),TRSPT);
3160:      GO((1,2,3,4,5,6),(-16.4, -679.7, 164.1+ZKORR, -43.9, 0.37, 87.7));
3170:      -- OLLAAN EDESSA YLHAALLA
3180:      GO((1,2,3,4,5,6),(-18.00,-679.7, 28.88,-44.00, 0.370, 88.0));
3190:      SPEED(SAKT);
3200:      GO(2,-667.72);
3210:      -- NYT OVAT JOHTIMIEN PAAT SISALLA
3220:      GO((1,3,5),(-22.04, 27.29, 1.406));
3230:      GO(2,-665.08);
3240:      -- NYT ON TAPPI SISALLA
3250:      GO(3,52.01+ZKORR);
3260:      SPEED(H);
3270:      GO(7,79);
3280:      GO(3,125+ZKORR);
3290: END;

```

```

3300: -----
3310: TASPATR7:SUBR;
3320: --          X          Y          Z          ROLL PITCH YAW GRIPO GRIPS
3330: TOKO(2.32, 213.52, -44.48, -103.4, 16.88, 0.0, 66.0, 48.00);
3340: END;
3350: -----
3360: PLASPATR7:SUBR;
3370: SPEED(0.15);
3380: GO(5,0); --RATA UT PITCHEN
3390: PLAKO(213.00, -449.70, 95.00+ZKORR, -104.92, 0, 0, 50.57, 66.0);
3400: SPEED(H);
3410: END;
3420: -----
3430: TAKONTR8:SUBR;
3440: --          X          Y          Z          ROLL PITCH YAW GRIPO GRIPS
3450: TOKO(-7.53, 70.91, -46.20, -43.50, 15.719, 0, 73.0, 48.0);
3460: END;
3470: -----
3480: PLAKONTR8:SUBR;
3490: GO((3,5),(175,0)); --RATA PITCHEN & HOJ
3500: GO((1,2, 4,5,6),(-21,-592, -133, -0.6, 87)); --OVER ANDRA PARTS
3510: PLAKO(-23.87, -595.83, 62.50, -133.53, 1.0346, 87.18, 48, 73);
3520: END;
3530: -----
3540: TAPLAT:SUBR;
3550: --          X          Y          Z          ROLL PITCH YAW GRIPO GRIPS
3560: TOKO(155.29, 216.47, -126.04, 43.58, -1.1064, 0, 30, 4.5);
3570: END;
3580: -----
3590: PLAPLAT:SUBR;
3600: --          X          Y          Z          ROLL PITCH YAW GRIPO GRIPS
3610: PLAKO(178.82, -560.48, 13.11+ZKORR, -45.344, -.922, 0, 4.5, 30);
3620: END;
3630: -----
3640: TAKABRA:SUBR;
3650: --          X          Y          Z          ROLL PITCH YAW GRIPO GRIPS
3660: TAKO(-61.59, 393.82, -70.59, -44.49, 19.425, 0, 39, 23.0);
3670: END;
3680: -----
3690: PLAKABRA:SUBR;
3700: --          X          Y          Z          ROLL PITCH YAW GRIPO GRIPS
3710: PLAKO(158.00, -516.34, 45.37+ZKORR, 45.22, -.922, 0, 23.0, 39);
3720: END;
3730: -----
3740: TAKOPL:SUBR;
3750: --          X          Y          Z          ROLL PITCH YAW GO GS
3760: SYSGRAS( KOPX, 355.70, -105.20, -134.08, -0.672, 90.211, 35.0, 1.8)
3770: IF KOPX GE 164 THEN KOPX=129.7 ELSE KOPX=164.1;
3780: END;
3790: -----
3800: RETKOPL:SUBR;
3810: --          X          Y          Z          ROLL PITCH YAW GO GS
3820: PLAKO(KOPX, 355.70, -105.40, -134.08, -0.672, 90.211, 1.8, 17);
3830: END;

```



```

3840: -----
3850: PLAKOPL: SUBR;
3860:   GO((1,2,3,5),TRSPT);
3870:   GO((1,2,3),(<177, -643, 179+ZKORR)); -- YLAFUOLELLA
3880:   GO((2,4,5,6),(<-734.01, -135.48,28.86,90.00)); -- PITCH:AUS
3890:   GO((1,3),(<161.17, 20.24+ZKORR)); -- KISKON PAALLA
3900:   SPEED(SAKT);
3910:   GDLTA((2,3,5),(< 3.58, 1.92, -0.68)); -- SOVITUS ALKAA
3920:   GDLTA((2,3,5),(< 3.68, 2.14, -1.07));
3930:   GDLTA((2,3,5),(< 6.39, 2.03, -1.19));
3940:   GDLTA((2,3,5),(< 4.46, 2.69, -0.81));
3950:   GDLTA((2,3,5),(< 5.42, 1.88, -0.99));
3960:   GDLTA((2,3,5),(< 6.44, 2.86, -1.32));
3970:   GDLTA((2,3,5),(< 6.94, 1.81, -1.28));
3980:   GDLTA((2,3,5),(< 6.57, 2.47, -1.51));
3990:   GDLTA((2,3,5),(<16.37, 3.88, -2.55));
4000:   GDLTA((2,3,5),(<15.00, 4.72, -2.21));
4010:   GDLTA((2,3,5),(<21.62, 4.32, -4.07));
4020:   GDLTA((2,3,5),(<13.38, 1.41, -4.17));
4030:   GDLTA((2,3,5),(<16.78, 2.12, -3.05));
4040:   GDLTA((2,3,5),(<25.02, -2.20, -3.37)); -- RUNTIAUS
4050:   DELAY(0.5);
4060:   -- PAIKALLA VIIMEISTAAN NYT
4070:   SPEED(H);
4080:   GO(3,200); -- YLOS
4090:   GO((1,2,3,5),TRSPT);
4100: END; -- PLAKOPL
4110: -----
4120: TAKO:SUBR(A,B,C,D,E,F,G1,G2);
4130:   GO(ARM,((A+80*SIN(E)),B,(C+80*COS(E)),D,E,F,G1));
4140:   SPEED(SAKT);
4150:   GO((1,3),(<A,C));
4160:   SPEED(H);
4170:   GO(7,G2);
4180:   GO((1,3),((A+150*SIN(E)),<(C+150*COS(E))));
4190: END;
4200: -----
4210: SSTA:NEW REAL; -- SENS VARIABEL
4220: KLTA:STATIC 2000.; -- KLAMKRAFTEN VID GRIPA
4230: -----
4240: TOKO:SUBR(A,B,C,D,E,F,G1,G2); -- TAKO MED SENSNING
4250: TOK1: GO(ARM,((A+80*SIN(E)),B,(C+80*COS(E)),D,E,F,G1));
4260:   SPEED(SAKT);
4270:   GO((1,3),(<A,C));
4280:   SSTA=SENSIO(SRP,REAL); -- SENS FORCE
4290:   SPEED(H);
4300:   GO(7,G2);
4310:   GO((1,3),((A+100*SIN(E)),<(C+100*COS(E))));
4320:   IF ABS(SSTA-SENSIO(SRP,REAL)) LT KLTA THEN
4330:     BEGIN BREAK('DETALJ SAKNAS. FIXA O TRYCK RETURN',EOL);
4340:     BRANCH(TOK1);END;
4350: END;
4360: -----
4370: PLAKO:SUBR(A,B,C,D,E,F,G1,G2);
4380:   IF G2 LT 72 THEN GO((1,2,3,5),TRSPT); --KOURAN VAPAA-ASEMA
4390:   GO(ARM,(<A,B,<(C+50),D,E,F,G1));
4400:   SPEED(SAKT);
4410:   GO(3,C);
4420:   SPEED(H);
4430:   GO(7,G2);
4440:   DELAY(.5);
4450:   GO(3,<(C+100));
4460: END;

```



```

10: --FILE:'WARTSTSK' UPPL 3/9-84 A.R. SOM EN DEL AV WART
20: ---*****
30: ---*
40: ---*
50: ---*
60: ---*
70: ---*
80: ---*
90: ---*
100: ---*
110: ---*
120: ---*****
130: --
140: DISPLAY('NU LADDAS WARTSTSK=SKRUVNING WARTSILA',EOL);
150: H:STATIC .10; -- DEFAULT HASTIGHET AR .10
160: SAKT:STATIC .05; -- LANGSAMMA RORELSE AR .05
170: -----
180: SKRUVHAL1:STATIC 4 OF 6 OF REAL IS(
190: -- HALLARE1 HALLARE2 HALLARE3 X-VARDEN
200: < 118.21, 88.50, 68.26, 39.31, 18.49, -9.69 >,
210: -- ! ! ! ! ! Y-VARDEN
220: < -435.72, -466.89, -436.47, -467.60, -436.81, -467.13 >,
230: -- ! ! ! ! ! DIR : 0-8
240: < 4, 8, 4, 8, 4, 8 >,
250: -- ! ! ! ! ! SKRUVANSATTNING
260: < 98.2, 98.2, 98.2, 98.2, 98.2, 98.2 >);
270: -----
280: SKRUVHAL2:STATIC 4 OF 8 OF REAL IS(
290: -- TRAF01 KONTEL4 PLAT KABRA X-VARDEN
300: < -45.03, -85.02, -1.61, 67.49, 195.73, 97.33, 176.72, 29.27 >,
310: -- ! ! ! ! ! Y-VARDEN
320: < -455.48, -499.40, -606.94, -606.73, -582.00, -583.65, -518.45, -518.96 >,
330: -- ! ! ! ! ! DIR : 0-8
340: < 3, 7, 1, 5, 0, 0, 0, 0 >,
350: -- ! ! ! ! ! SKRUVANSATTNING
360: < 87.1, 87.1, 90.7, 90.7, 85.8, 85.8, 87.4, 87.4 >);
370: -----
380: SKRUVHAL3:STATIC 4 OF 7 OF REAL IS(
390: -- SPATR !! KONTR8 X-VARDEN
400: < 206.88, 207.70, 142.30, 142.56, -49.01, -26.60, -66.33 >,
410: -- ! ! ! ! ! Y-VARDEN
420: < -417.14, -478.68, -418.25, -480.00, -539.91, -644.20, -644.22 >,
430: -- ! ! ! ! ! DIR : 0-8
440: < 4, 6, 2, 8, 3, 7, 7 >,
450: -- ! ! ! ! ! SKRUVANSATTNING
460: < 88.9, 88.9, 88.9, 88.9, 87.3, 87.3, 87.3 >);
470: -----
480: TASKRUVDR:SUBR;
490: -- X Y Z ROLL PITCH YAW GO GS
500: SYSGRAS( 193.63, -200.73, -34.78, -45.00, -.947, 0, 35, 7 );
501: GO(4,5,6),(-45.0, -1.288, -2.143)); -- STABIILI ASENTO
510: END;
520: -----
530: LASKRUVDR:SUBR;
540: -- X Y Z ROLL PITCH YAW GO GS
550: SYSGRIP( 195.00, -200.73, -34.58, -45.37, -.9471, 0, 7, 35 );
560: END;

```

```

570: -----
580: SKRUVNING:SUBR;
590:   PA('S3');
600:   SENSIO(VVAC,ON); -- START VACUUMPUMP
610:   SENSIO(D4,ON); -- OFFNA VENTILEN
620:   SENSIO(B4,ON); -- SLA PA TRYCKLUFT
630:   SKX=SKKONSTX; -- NOLLATAAN RUUVILASKURI
640:   SKY=SKKONSTY;
650:   MAP($SKRV,SKRUVHAL1);
660:   IF SKY GT -150.0 THEN UPDAT; -- PIENET RUUVIT
670:   MAP($SKRV,SKRUVHAL2); -- TRAF01 & KONTEL & PLAT
680:   SENSIO((B4,D4),OFF); -- STANG AV
690:   SENSIO(VVAC,OFF);
700:   AV('S3');
710: END;
720: -----
730: SKRUVNING2:SUBR;
740:   PA('S3');
750:   SENSIO(VVAC,ON); -- START VACUUMPUMP
760:   SENSIO(D4,ON); -- OFFNA VENTILEN
770:   SENSIO(B4,ON); -- SLA PA TRYCKLUFT
780:   MAP($SKRV,SKRUVHAL3);
790:   SENSIO((B4,D4),OFF); -- STANG AV
800:   SENSIO(VVAC,OFF);
810:   AV('S3');
820: END;
830: -----
840: SKRV:SUBR(A,B,DIR,SKRVANS); -- KOMPONENTTIEN RUUVAUS
850:   SKRMON: NEW MONITOR(SLT,3,-10000,3500);
860:   ERRCNT2: NEW 0;
870:   ERRCNT3: NEW 0;
880:   ERRCNT4: NEW 0;
890: --
900: ROSIDA: SUBR;
910:   IF DIR EQ 1 THEN
920:     GO((1,2),((A-15),B));
930:   ELSE IF DIR EQ 2 THEN
940:     GO((1,2),((A-10),(B+10)));
950:   ELSE IF DIR EQ 3 THEN
960:     GO((1,2),((A),(B+15)));
970:   ELSE IF DIR EQ 4 THEN
980:     GO((1,2),((A+10),(B+10)));
990:   ELSE IF DIR EQ 5 THEN
1000:     GO((1,2),((A+15),B));
1010:   ELSE IF DIR EQ 6 THEN
1020:     GO((1,2),((A+10),(B-10)));
1030:   ELSE IF DIR EQ 7 THEN
1040:     GO((1,2),((A),(B-15)));
1050:   ELSE IF DIR EQ 8 THEN
1060:     GO((1,2),((A-10),(B-10)));
1070:   ELSE GO((1,2),((A,B)));
1080: END; --ROSIDA

```

```

1090: --
1100: SK2:
1110: GO((1,2,3),TRSPT2); -- ALUSTAN JA RUUVIMAKASIININ VALIIN
1120: IF TASKRUV NE 0 THEN
1130: BEGIN
1140:   RAPO('VIKA RUUVIMAKASIINISSA ');
1150:   ERRCNT2=10;
1160:   BRANCH(ENDSKRH);
1170: END;
1180: GO((1,2,3),TRSPT2);
1190: -- RUUVIN REIAN VIEREEN
1200: SPEED(H);
1210: ROSIDA;
1230: GO(3,(SKRVANS+20));
1240: -- RUUVIN REIAN PAALLE
1250: SPEED(.02);
1260: GO((1,2),(A,B));
1270: DELAY(0.5);
1280: IF NOT SENSIO(C11,0) THEN -- TAPPAT SKRUV, TA NY
1290: BEGIN
1300:   RAPO('RUUVI FUDONNUT');
1310:   ERRCNT2=ERRCNT2+1;
1320:   --
1330:   SPEED(H);
1340:   GO(3,215);
1350:   IF ERRCNT2 GT 2 THEN BRANCH(ENDSKRH);
1360:   BRANCH(SK2);
1370: END;
1380: ERRCNT2=0;
1390: -- RUUVI ON REIAN PAALLA OIKEIN
1400: REMONITOR(SKRMON);
1410: GO(JZ,(SKRVANS+1.5),SKRMON); -- RUUVI ON REIAN PAALLA
1420: -- RUUVI AUSAUS ALKAA
1430: SPEED(.005);
1440: GDLTA(JZ,-14,SKRMON);
1450: GO(3,SKRVANS-5,SKRMON);
1460: IF NOT SENSIO(C11,0) THEN -- ALA-ASENNON ALIPAINEN
1470: BEGIN
1480:   ERRCNT3=ERRCNT3+1;
1490:   RAPO('RUUVIAUKSESSA HAIRIO');
1500:   --
1510:   SPEED(H);
1520:   GO(3,215);
1530:   IF ERRCNT3 GT 2 THEN BRANCH(ENDSKRH);
1540:   BRANCH(SK2);
1550: END;
1560: ERRCNT3=0;
1570: -- VAHAN (0.5 MM) IRTI KANNASTA
1580: SPEED(H);
1590: GO(3,SKRVANS+2.5,SKRMON); -- BACKA LITE
1600: DELAY(1.0);
1610: IF SENSIO(C11,0) THEN -- RUUVI LOYSALLA
1620: BEGIN;
1630:   GO(3,SKRVANS+10,SKRMON);
1640:   -- RUUVI AUSAUS UUELLEEN
1650:   SPEED(0.005);
1660:   REMONITOR(SKRMON);
1670:   GO(JZ,SKRVANS-13,SKRMON);
1680: END;
1690: -- NOUSTAA YLOS, RUUVI EI SAA SEURATA
1700: GO(3,SKRVANS+5);
1710: SPEED(H);
1720: GO(3,210);
1730: DELAY(0.5);

```



```

1720: IF SENSIO(C11,0) THEN -- RUUVI SEURAA
1730: BEGIN
1740: RAPO('RUUVI EI RUUVAUTUNUT');
1750: ERRCNT4=ERRCNT4+1;
1760: GO(3,210); RUUVI EI OLE KIERTYNYT
1770: IF ERRCNT4 GT 2 THEN BRANCH(ENDSKRH);
1780: BRANCH(SK2); -- SKRUV FASTNADE EJ, SPOTTA UT OCH TAG NY
1800: END;
1810: ERRCNT4=0;
1820: --
1830: ENDSKRH:
1840: ENDMONITOR(SKRMON);
1850: GO(3,210);
1860: GO((1,2,3),TRSP2);
1870: --
1880: IF (ERRCNT2+ERRCNT3+ERRCNT4) GT 0 THEN
1890: BEGIN
1900: RAPO('*** RUUVI AUS KESKEYTYNYT!');
1910: RETURN(10);
1920: END; VIRHEILMOITUS, KUN RUUVI JAA PUUTTUMAAN
1930: RETURN(0);
1940: END; --SKRV
1950: -----
1960: SKKONSTX:STATIC 8.63; -- ATERSTALLNINGSKONSTANT X
1970: SKKONSTY:STATIC -144.94; -- " " Y
1980: SKX:STATIC SKKONSTX; -- SKRUVMAGASINET X-VARDE
1990: SKY:STATIC SKKONSTY; -- " " Y-VARDE
2000: SKDELX:STATIC 9.906; -- DELNING I X I MAGASINET
2010: SKDELY:STATIC 10.268; -- " " Y " "
2020: -----
2030: TASKRUV:SUBR; -- TASKRUVEN
2040: -- X Y Z ROLL PITCH YAW GS
2050: RETURN(TASK(SKX, SKY, -29.20, -45, -1.288,-2.143, 9.86));
2060: END;
2070: -----
2080: TASK:SUBR(A,B,C,D,E,F,G1); -- TA SKRUV
2090: SPEED(H);
2100: ERRCNT1: NEW 0; OHJELMA HAKEE UUDEN RUUVIN TELINEESTÄ
2110: TS1:
2120: GO((1,2,3),SPOTPLATS); -- POISTETAAN MAHDOLLINEN VANHA RUUVI
2130: SENSIO(D4,OFF);
2140: DELAY(1.0);
2150: SENSIO(D4,ON);
2160: GO(ARM,(A,B,(C+60),D,E,F,G1));
2170: GO(3,(C+3));
2180: SPEED(SAKT);
2190: GO(3,C); -- GA NER OCH SUG UPP SKRUV
2200: DELAY(.3);
2210: SPEED(H);
2220: GO(3,(C+30)); -- BACKA UPP
2230: SKX=SKX-SKDELX;
2240: IF SKX LT -75 THEN UPDAT;
2250: DELAY(0.5);
2260: IF NOT SENSIO(C11,0) THEN -- VINOSSA TAI EI RUUVIA
2270: BEGIN
2280: ERRCNT1=ERRCNT1+1;
2290: IF ERRCNT1 GT 2 THEN RETURN(101);
2300: A=SKX; -- PAIVITETAAN MYOS X-KOORD
2310: B=SKY; -- PAIVITETAAN Y-KOORD
2320: RAPO('RUUVIN HAKU EPAONNISTUI');
2330: BRANCH(TS1); -- MISSADE SKRUVEN,SPOTTA,GA TILL NASTA,TA N
2340: END;
2350: RETURN(0);
2360: END;
2370: -----
2380: UPDAT:SUBR; -- UPDAT AV LAGEN I SKRUVMAGASINET
2390: SKY=SKY-SKDELY;
2400: IF SKY LT -167 THEN SKY=SKKONSTY-SKDELY; -- TOISELLE RIVILLE
2410: SKX=SKKONSTX;
2420: END;

```

LIITE D. KÄYTTÄJÄLIITYNNÄN TULOSTEITA

```
*****  
***                                     ***  
***           WARTSILA   ROBOTICS           ***  
***                                     ***  
*****
```

OHJELMASSA KAYTETAVA KIELI/
SPRAKET, SOM SKALL ANVANDAS I PROGRAMMET

0 SVENSKA (SWE)
1 SUOMI (FIN)

VALINTA/VAL (1 NUM) : 1

```
*****
***                                     ***
***           WARTSILA   ROBOTICS           ***
***                                     ***
*****
```

ROBOTTISOVELLUTUKSET :

```
0 ..... LOPETUS
1 ..... KELLON ASETUS
2 ..... TUOTE A:N (DEMO) VALMISTUS
3 ..... (EI KAYTÖSSÄ)
```

VALINTA (1 NUM) :

*** WARTSILA ROBOTICS ***

TUOTTEEN A VALMISTUSTA EDELTAVAT TOIMET:

- TARKISTA, ETTA KYTKENTA_ALUSTA ON OIKEIN PAIN
- TARKISTA, ETTA ROBOTIN KOURA ON TYHJA.
- PANOSTA MAKASIINIT SITEN ETTA KOMPONENTIT
OVAT SEURAAVASSA JARJESTYKSESSA OIKEALTA VASEMMALLE

- | | |
|--------------------------------------|-----------------------|
| 1. VIRTAMUUNTAJA | 6. LAMPORELE |
| 2. 3 KPL SULAKEPESIA | 7. MUUNTAJA |
| 3. KONTAKTORIN APUKYTKIN | 8. KONTAKTORI (SUURI) |
| 4. EROTIN | |
| 5. KAAPELIKOURU / KONTAKTORI (PIENI) | |

- TARKISTA, ETTA RUUVEJA ON RIITTAVASTI

LOPUKSI PAINA "ENTER"-NAPPAINTA !

***** WARTSILA ROBOTICS *****

JARJESTELMAKELLON ASETUS

UUSI AIKA KIRJOITETAAN VANHAN AJAN
PAALLE MUODOSSA:

<TUNTI, MIN, SEK, MSEK, KUUKAUSI, PAIVA, VUOSI>

ESIM. <08,14,30,000,04,01,84>

AJAN ASETUS : <10,54,38,390,9,14,84>

8.8.56	==>	KALUSTUS ALKAA
		NOPEUDET OVAT .800000 / .0500000
8.15.41	==>	RUUVI EI RUUVAUTUNUT
8.18.35	==>	ALUSTA KALUSTETTU
		KALUSTUS KESTI 585.578 s

8.25.38	==>	KALUSTUS ALKAA
		NOPEUDET OVAT .800000 / .0500000
8.35.11	==>	ALUSTA KALUSTETTU
		KALUSTUS KESTI 578.262 s

KAHDEN ALUSTAN VALMISTUKSESTA TULLEET RAPORTIT.